

2次元スキャンデータに基づく環境地図構築・自己位置推定 のための特徴量生成に関する研究

令和元年 9 月

和歌山大学大学院システム工学研究科

脇田 翔平

Feature Generation for Map Construction and Self-Localization based on 2D Scan Data

September 2019

Graduate School of Systems Engineering
Wakayama University

Shohei Wakita

概要

自律型移動ロボット普及のためには、自動的な環境地図構築技術や大域的自己位置推定技術が必要不可欠である。大域的自己位置推定技術では、ロボットの初期の位置姿勢を与えずに、ロボットの地図内における位置姿勢を推定しなければならない。さらに、地図作成時と環境が変化した場合、意図せずに自分の位置を見失った場合や別の場所に移動された場合などに、正確に大域的自己位置推定できる能力を持つことが望まれる。そこで、2次元レーザスキャナを用いて大域的自己位置推定を実現する際には、地図データ内の対応する箇所とスキャン（2次元レーザスキャナで取得した1時刻のデータ）の位置合わせを行うスキャンマッチングがよく用いられる。スキャンマッチングでは、スキャンから環境の形状を表す特徴量を抽出し、それを利用して地図との位置合わせを行う手法が数多く提案されてきた。特徴量を利用するスキャンマッチングによる位置合わせが困難になる代表的な状況として、スキャンにノイズやオクルージョンが発生している場合、環境中に移動物体が存在する場合などがあり、これらは同時に発生する可能性もある。これらが発生した場合は、スキャンの位置合わせにおいて有用な形状特徴が不足し、結果として自己位置推定に失敗してしまう場合がある。

本研究では、自己位置推定を頑健に行うために、2つのアプローチを提案する。一つ目のアプローチとして、スキャン内に位置合わせに有用な形状特徴が不足する場合は局所地図を生成することで、形状特徴の不足を補い、生成した局所地図と環境地図とでスキャンマッチングを行い、自己位置推定する手法を提案する。スキャン内の形状特徴の過不足を判断するために特徴量“地形複雑度”を提案し、この特徴量に基づいて局所地図を作成する方法“選択的統合処理”を提案する。形状特徴が十分に含まれた局所地図を用いることで、我々の研究グループで過去に提案した CIF-based スキャンマッチング法よりも、大域的自己位置推定がより頑健に行えるようになる。実験では、実環境での大域的自己位置推定を行い、「ロボット誘拐問題」の状況を想定した Monte Carlo Localization (MCL) 法を用いた結果と比較して、提案手法の有効性を示した。二つ目のアプローチとして、移動ロボットの走行環境における自己位置推定に適した特徴量を自動で生成するために、variational autoencoder(VAE) を独自に拡張した LaserVAE 法と、LaserVAE によって生成した特徴量を利用した自己位置推定する手法を提案する。LaserVAE は、スキャンを圧縮して特徴量を生成する機能と、生成した特徴量からスキャンの再構成を行う機能を有するマルチタスクニューラルネットワークとなっている。LaserVAE によって生成した特徴量のみを環境地図として保持することで、生のスキャンを保持する必要がなく、地図のデータ量を削減できるという特徴を持つ。実験では、実際の屋内・屋外環境において2次元レーザスキャナを用いた大域的自己位置推定の実験を通して、LaserVAE の有効性を示した。

Abstract

In order to spread autonomous mobile robots, automatic map construction technology and global self-localization technology are indispensable. For the global self-localization technology, it is necessary to estimate the position and orientation in the map of the robot without giving the initial position and orientation of the robot. Furthermore, it is desirable for the self-localization technology to have the ability to be able to estimate global self-position accurately in case that the environment is not static, or that the robot loses its own position and orientation due to a sensor system failure, or that the robot is instantly moved to other position without being told during the operation of the robot. In order to achieve global self-localization using a 2D laser scanner, scan matching is often used to align a scan acquired by the 2D laser scanner at a time with the map for finding the corresponding location. A number of scan matching methods have been proposed and most of them extract feature descriptors representing the shape of the environment from the scan and align such descriptors in the scan with the descriptors in the map. A scan is often affected by sensor noises, occlusion and/or moving objects in the environment. Because it is hard to extract useful feature descriptors of the environment under such situations, scan matching methods using feature descriptors tend to fail correct estimation.

This study proposes two approaches for performing robust and correct self-localization using 2D scan data. One is a selective merging method of 2D range scans for enhancing our CIF-based scan matching algorithm which is a global scan matching algorithm using the CIF descriptors and a geometric constraint between keypoints. The selective merging method can generate a local map based on “terrain complexity” without fixing the number of scans which are used to make a local map. Through experiment in real environment, we show that our enhanced version of the CIF-based scan matching algorithm can perform global scan matching more robustly in a large cluttered environments without using an initial alignment. The second is “LaserVAE” that is an enhanced version of the state-of-the-art variational auto-encoder (VAE) by introducing the step-edge detector, which detects non-continuous transition emerged frequently at the laser scan data due to the limitation of distance measurement. With our proposed method, the feature descriptor of the laser scan is automatically tuned given unknown environments. Through experiments with a real self-localization with a 2D laser scanner, we demonstrate the effectiveness of “LaserVAE.”

目次

1	序論	1
1.1	研究背景	1
1.2	本論文で対象とする環境と移動ロボット	3
1.3	選択的統合処理に基づく CIF ベース大域的自己位置推定へのアプローチ	5
1.4	LaserVAE による特徴量生成とその特徴量に基づいた大域自己位置推定へのアプローチ	6
1.5	関連研究	7
1.5.1	大域自己位置推定に関する様々なアプローチ	7
1.5.2	スキャンマッチング法	7
1.5.3	スキャンから特徴量を直接生成する方法	9
1.5.4	スキャンを画像化して特徴量を生成する方法	10
1.5.5	環境地図圧縮法	10
1.6	本論文の構成	12
2	選択的統合処理に基づく CIF ベース大域的自己位置推定	13
2.1	前処理と地形複雑度の算出	14
2.2	選択的統合処理	17
2.3	CIF-based スキャンマッチング	21
2.3.1	特徴量の算出	22
2.3.2	対応点探索	25
2.3.3	スキャンマッチング	27
2.4	検証実験	29
2.4.1	地形複雑度によるスキャンの形状特徴の判断の検証	29
2.4.2	選択的統合処理の検証	30
2.4.3	提案手法による大域的自己位置推定の検証	31
2.4.4	未学習な環境における提案手法の有効性の検証	36
2.5	まとめ	39
3	LaserVAE による特徴量生成とその特徴量に基づいた大域自己位置推定	40

3.1	LaserVAE	41
3.1.1	LaserVAE のネットワーク構造	41
3.1.2	LaserVAE の学習方法	43
3.1.3	LaserVAE を用いた特徴量生成とスキヤンの再構成	45
3.2	大域的自己位置推定	46
3.3	検証実験	48
3.3.1	特徴量の回転不変性に関する検証	48
3.3.2	特徴量からのスキヤンの再構成の性能	50
3.3.3	大域的自己位置推定の性能	51
3.3.4	環境地図のデータ量圧縮の性能	54
3.4	まとめ	55
4	結論	57
	謝辞	61
	参考文献	62
	研究業績	65

1 序論

1.1 研究背景

近年，お掃除ロボットや自動車の自動運転など，移動ロボットによる環境の自律走行技術が注目を集めている．移動ロボットが自律的に移動するためには，走行する環境を事前に地図として保持しておく必要がある．また，走行時には地図における現在の自己位置を知る必要がある．大域的自己位置推定技術では，ロボットの初期の位置姿勢を与えずに，ロボットの地図内における位置姿勢を推定しなければならない．さらに，地図作成時と環境が変化した場合，意図せずに自分の位置を見失った場合や別の場所に移動された場合などに，正確に大域的自己位置推定できる能力を持つことが望まれる．

大域的な自己位置推定では，ロボットに取り付けられたセンサで取得した環境情報と地図情報を照合し，地図内においてロボットがどこに位置しているのかを発見しなければならない．

移動ロボットに取り付けるセンサとしてカメラやレーザスキャナ（またそれらを組み合わせたもの）がよく用いられる．カメラは環境のテクスチャ情報を密に取得できるが，ステレオカメラを用いたとしても奥行き情報の信頼性はレーザスキャナに劣る．対してレーザスキャナはレーザ光の跳ね返り時間によってデータを取得することで奥行き情報を精度よく取得可能であるが，レーザの回転によって周囲の距離情報を取得するため，レーザスキャナから取得できるデータは近い地点の点群は密になり，遠くなるほど粗になる．

近年は，前述した自動運転の研究が盛んに行われており，屋外環境においても奥行き情報を精度よく取得できる 3D レーザスキャナを利用した自己位置推定に関する研究が多く行われている [1, 2]．3D レーザスキャナは走行環境が不整地の場合でも環境情報を取得できるというメリットがある．一方，我々人間の生活圏には，平坦な路面や建物の壁面で構成されるような整地された環境が存在する（学校，工場，公園等）．3D レーザスキャナはカメラや 2D レーザスキャナに比べて高価であり，屋内の整地環境のみを走行する場合や，屋外であっても舗装された路面を走行する場合などは 2D レーザスキャナのみで自己位置推定を行うことは十分可能である．

本研究では，このような整地された環境において，2D レーザスキャナを用いて取得したスキャン（センサの周囲一周分の距離データ）から環境の形状特徴を抽出する方法及びその特徴を用いた大域的自己位置推定を行う手法を提案する．

2D レーザスキャナを用いて自己位置推定を行う方法の一つにスキャンマッチングを利用する方法がある．スキャンマッチングとは，二つのスキャンの形状を重ね合わせ，スキャン間の位置姿勢の変化量

を推定する方法である．スキャンマッチングでは，レーザスキャナで取得したスキャンをそのまま用いて位置合わせする方法 [3, 4, 5] がある．この方法では，スキャン内の移動物体等のノイズにより位置合わせが不安定になる場合がある．そのため，スキャンから環境の形状を表す特徴量を抽出し，その特徴量同士を対応づけて位置合わせする方法 [6, 7, 8] が提案されてきた．特徴量とは，入力データを固有の数値表現に変換したもので，スキャン内から環境の形状情報 (直線分，コーナ点，ステップエッジ等) を表す固有の数値を取得する．

特徴量を用いたスキャンマッチングでは，スキャンから地形の形状を表す特徴量を抽出するために，コーナ等の特異な点を抽出し，その特異な点周辺の点群の分布等を特徴量として利用するものがある．

問題 1: 2D レーザスキャナで取得できる点群データ内に形状特徴が十分に存在しない場合，自己位置推定が不安定になる問題

スキャンマッチングによる位置合わせが困難になる代表的な状況として，スキャンにノイズやオクルージョンが発生している場合，環境中に移動物体が存在する場合などがあり，これらは同時に発生する可能性もある．これらが発生した場合は，スキャンの位置合わせにおいて有用な形状特徴が不足し，結果として自己位置推定に失敗してしまう場合がある．

本研究では，問題 1 に対するアプローチとして“選択的統合処理に基づく CIF ベース大域的自己位置推定”を提案する (提案手法 1)．この方法により，スキャン内に形状特徴が少ない場合であっても局所地図を作成し形状特徴を増加させることで，位置合わせを行えるようになることを示した．

問題 2: 走行環境によって取得できる形状特徴が異なる問題

スキャンマッチングを行う場合，スキャンから環境の形状を表す特徴量を抽出し，その特徴量同士を対応付けることでスキャン間の位置合わせを行う手法がある．このような特徴量は人手によって設計されており，適用できる環境が限られる場合が多い (直線分で構成される環境や，静的な環境)．一方，ロボットの走行環境は様々である．そこで，ロボットの走行環境に合った特徴量を自動で構築することが望まれる．

問題 2 に対するアプローチとして“LaserVAE による特徴量生成とその特徴量に基づいた大域自己位置推定”を提案する (提案手法 2)．この方法では，ロボットの走行する環境のスキャンをニューラルネットワークで学習し，特徴量生成を自動で行うことを提案する．実験により，スキャンから屋内や屋外の動的環境にも対応した特徴量が生成可能であることを示す．

1.2 本論文で対象とする環境と移動ロボット

本研究で想定する走行環境について述べる．また，移動ロボットとそれに搭載するセンサについて述べる．提案手法 1 では，オフィス内を走行する移動ロボットを想定し，和歌山大学システム工学部棟 (Fig. 1) において実験を走行実験を行った．

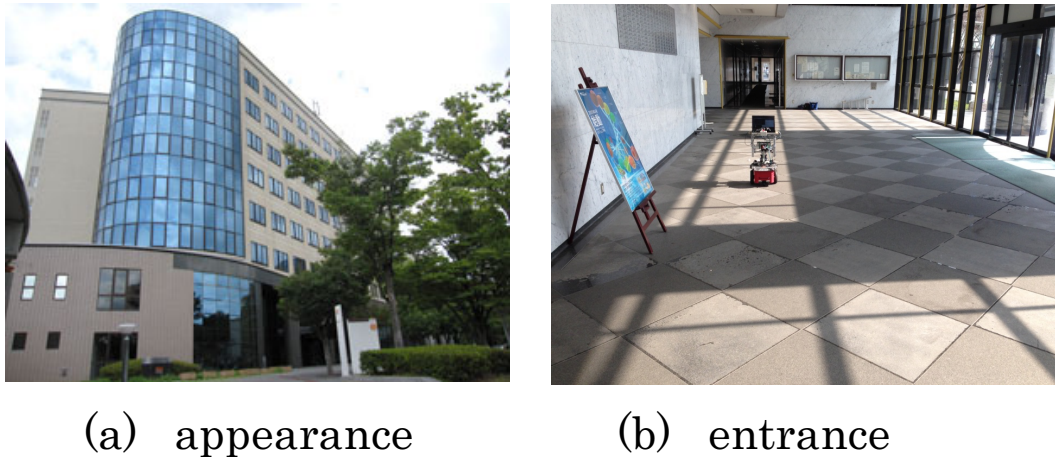


Fig. 1 Appearance and the entrance of Faculty of System Engineering Building

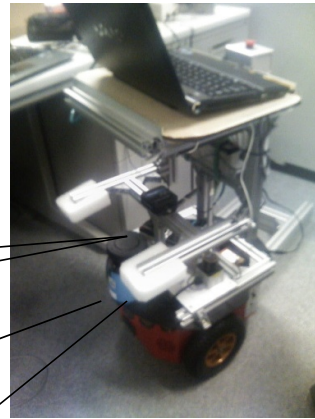
提案手法 2 では，屋内・屋外を問わないあらゆる整地環境での走行を想定している．また，人の往来のある動的な環境での走行も想定している．実験環境は，屋外環境として和歌山大学のシンボルゾーンと，屋内環境に和歌山大学システム工学部 1 階で走行実験を行った．

本論文で対象とするロボットについて述べる．提案手法 1 では主に会社のオフィスや大学構内等の整地された屋内環境を走行するロボットを想定している．本研究ではリバスト社の“PIONEER 3-DX”に SICK 社のレーザスキャナを搭載し使用した (Fig. 2)．

提案手法 2 では，屋内外問わず走行可能な四輪移動ロボット“mercury” (株式会社リバスト) に 2 次元レーザスキャナ (SICK 社, LMS-151, 視野角 270[deg], 角度分解能 0.5[deg]) を地上からの高さ約 60[cm] の位置に搭載し使用した (Fig.3)．

レーザスキャナについて述べる．提案手法ではレーザスキャナを用いて得られる距離データから，環境固有の特徴量を生成し，自己位置推定を行う．レーザスキャナの距離データの取得方法を図に示す．レーザを周囲に向けて照射しその跳ね返り時間によって距離を測定する Time Of Flight(TOF) 方式が用いられている．

SICK, Inc. LMS-100
 Aperture angle 270[deg]
 Angular resolution 0.25[deg]



OS: Ubuntu12.04LTS ,
 CPU:Corei7 620M
 robot : Pioneer 3-DX

Fig. 2 PIONEER 3-DX(REVAST .Co., Ltd.) with Laser Scanner(SICK, Inc.)

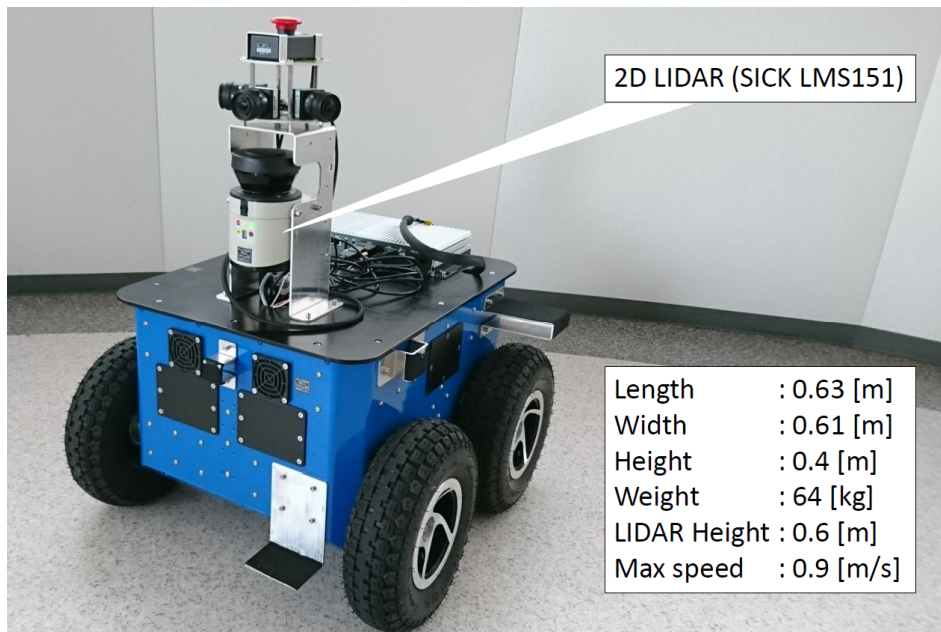


Fig. 3 Mobile robot "mercury" (REVAST .Co., Ltd.)

1.3 選択的統合処理に基づく CIF ベース大域的自己位置推定へのアプローチ

スキャンマッチング法には、位置合わせが困難になる状況が存在する（スキャンにノイズやオクルージョンが発生している場合、環境中に移動物体が存在する場合、スキャンの位置合わせにおいて有用な形状特徴が不足している場合など）。これらの問題を解決する方法として、複数時刻のスキャンを用いて局所地図を生成し、生成した局所地図と環境地図とでスキャンマッチングを行い、自己位置推定する方法がある。この場合は、連続して入力されるスキャンの中から局所地図生成に用いるスキャンをどのように選択するかが問題となる。局所地図生成に用いるスキャンの数を予め限定すると、生成された局所地図内に環境を表す形状特徴が不足する可能性があり、そのような状況では、正しく位置合わせを行うことが難しくなる。一方で、一時刻分のスキャンに形状特徴が豊富にある場合、複数時刻分のスキャンを用いて局所地図を生成する必要はない。大域的自己位置推定に用いるスキャン数を最小にするためには、環境内に含まれる形状特徴に応じて局所地図を生成するか否かを判断して、局所地図生成に用いるスキャンを適切に選択して位置合わせすることが望ましい。

そこで、本研究では、スキャンの形状特徴を表す特徴量“地形複雑度 (terrain complexity)”を提案し、この特徴量に基づいて局所地図を作成する方法“選択的統合処理”を提案する。“地形複雑度”は、スキャン内の環境の形状特徴を表し、それが十分に含まれているか否か（環境の形状特徴の有無）を判断することを、2 クラスの識別問題として捉え、何らかの識別器によってその判断を下す。“選択的統合処理”は、識別器の判断結果に基づき、用いるスキャンを適切に選択して局所地図を生成する手法である。この選択的統合処理により、用いるスキャンの数を予め決定することなく、形状特徴が十分に含まれた局所地図を生成できる。この局所地図を用いることで、我々の研究グループで提案した CIF-based スキャンマッチング法 [9, 10, 11] での大域的自己位置推定法がより頑健に行えるようになる。提案手法の有効性を検証するために、実環境の地図を用いて大域的自己位置推定を行う実験を行い、地形複雑度に基づいてスキャンに含まれる形状特徴が十分か否かを判断できることを示し、位置合わせに用いるスキャンの数を限定せずに局所地図を生成でき、その局所地図を用いて大域的自己位置を頑健に行えることを示す。また、いわゆる「ロボット誘拐問題」の状況を想定した Monte Carlo Localization (MCL) 法 [12] を用いた大域的自己位置推定の結果と比較して、提案手法の有効性を示す。

詳細は 2 章にて述べる。

1.4 LaserVAE による特徴量生成とその特徴量に基づいた大域自己位置推定へのアプローチ

従来からの手法では、環境を表す特徴量は人手によって設計されており、必ずしも環境に適した特徴量を生成できるとは限らない。実際にロボットを自律走行させる場合、走行する環境はさまざまであり、その環境毎に安定して生成できる特徴量が望まれている。

また、従来手法で用いられている地図には点群地図 [13] や占有格子地図 [12, 14, 15], 特徴地図 [16, 17, 18, 19, 20, 21] など様々なものがある。ロボットの行動範囲が大きくなれば、広い範囲の地図を用意する必要があり、地図を広くすればするほど、保持するためのメモリ量が増大する。

そこで、本研究では、移動ロボットの走行環境における自己位置推定に適した特徴量を自動で生成するために、variational autoencoder (VAE)[22] を独自に拡張した LaserVAE を提案する。また、LaserVAE によって生成した特徴量を利用した自己位置推定法を提案する。VAE は二つの機能 (encoder, decoder と呼ばれる) を持つニューラルネットワークである。encoder は入力データを圧縮する機能を持ち、decoder は圧縮されたデータから元の入力データを復元するという機能を持つ。ニューラルネットワークの学習後、encoder と decoder は別々に利用できる。VAE は潜在変数（入力データを圧縮したもの）をガウス分布で表現しており、学習後のネットワークを生成モデルとして活用できるため、近年、人物の顔画像のモンタージュ [23] や風景写真のスタイル変換 [24] などに応用され、注目を集めている。

我々の提案する LaserVAE は、通常の VAE にスキャン内のステップエッジ部分を検出する機能を有するネットワーク (step-edge classifier) を追加したもので、スキャンを圧縮して特徴量を生成する機能と、生成した特徴量からスキャンの再構成を行う機能を有するマルチタスクニューラルネットワークとなっている。LaserVAE は静的な環境だけでなく動的な環境においても使用可能な特徴量を生成することができる。また、特徴量のみを環境地図として保持することで、生のスキャンを保持する必要がなく、地図のデータ量を削減できるという特徴を持つ。本論文では、実際の屋内・屋外環境において 2 次元レーザスキャナを用いた大域的自己位置推定の実験を通して、提案する LaserVAE の有効性を示す。

詳細は 3 章にて述べる。

1.5 関連研究

大域的自己位置推定法は、位置姿勢の初期値を持たせず、ロボットの持つ外界センサから得られる情報と環境地図とを照合し、ロボットの位置姿勢を推定する。大域的自己位置推定法に関する研究例では、カメラやレーザスキャナといった外界センサが用いられることが多い。

1.5.1 大域自己位置推定に関する様々なアプローチ

Konolige ら [25] は、ロボットの行動範囲内全ての位置姿勢を一定間隔で量子化しておき、時々刻々のスキャンデータと地図との相互相関に基づく尤度を再計算して、環境地図内で自己位置を絞り込んで自己位置推定を行う Grid Localization を提案した。この手法は計算量が多く処理時間が長くなるという問題があったため、Olson ら [26] は、並列計算機の利用や位置姿勢の量子化を階層化し、高速化を図った。

Dellaert ら [12] は、Grid Localization と違い、環境地図を格子に区切ることなく、位置姿勢の候補を粒子（パーティクル）で表し、スキャンデータが入力されるたびに尤度の低い粒子を淘汰していくことで自己位置推定を行う Monte Carlo Localization (MCL) 法を提案している。

また、MCL 法においていわゆる「ロボット誘拐問題」が発生した場合の解消法として、Gutmann ら [27] は、誘拐発生時にパーティクルの分布を再計算し復帰を試みる方法を提案している。この方法では誘拐による移動先が近距離である場合にもパーティクルの分布を広げすぎてしまうため無駄が大きい。

それに対して上田ら [28] は、誘拐発生直前のパーティクルの情報を残しつつパーティクルの分布を広げて、復帰を試みる方法を提案している。

1.5.2 スキャンマッチング法

レーザスキャナから得られるスキャンの同士の位置合わせを行う手法をスキャンマッチングという。スキャンマッチング法を利用した大域的自己位置推定法として、スキャンから抽出した直線分と地図の直線分とを対応づけて自己位置推定を行うもの [6] や、スキャンからコーナ点やステップエッジ点やドアなどの特異な特徴点をスキャン点列から抽出し、それらの特徴点をノードとした完全グラフを構築して、各特徴点のマッチング結果を集計してスキャン間の大域的スキャンマッチングを行う方法 [7] などが存在する。

しかし、スキャンマッチング法は、どんな状況でも正しく位置合わせができるわけではない。スキャンの位置合わせが困難になる代表的な状況として、スキャンにノイズやオクルージョンが発生している場合、環境中に移動物体が存在する場合、スキャンの位置合わせにおいて有用な形状特徴が不足してい

る場合などがあり、これらは同時に発生する可能性もある。

これらの問題に対して、前処理によりノイズを除去するもの [29] や、オクルージョンによる特徴点の誤検出に対応するため、スキャン内の各スキャン点において、スキャン全体を表す特徴ベクトルを生成するもの [8] が提案されている。さらに、移動物体が存在する環境に対応するための手法として、移動物体に相当するスキャン内のデータを除去するもの [30] が提案されている。我々の研究グループでも、スキャンマッチングの一手法として、スキャン内のデータの欠損や変化があっても環境中から不変な箇所を探索し安定して地図を更新できるように、合同変換に不変な CIF 特徴量 (Congruence transformation Invariant Feature) [31] を用いたスキャンマッチング法 [9, 10] (以後, CIF-based スキャンマッチング法と呼ぶ) を提案した。CIF 特徴量は、キーポイント^{*1} の周囲の点の分布等をヒストグラムとすることで、キーポイント周辺の形状を表す特徴量を得る。これにより、スキャンの形状に変化があった場合でも、局所的に一致する箇所を探索して位置合わせを行うことができる。

スキャンにその環境を表す形状特徴が不足している場合に対する、1つの解決手法として、原ら [32] は、現在のスキャンのみを用いて位置合わせを行うのではなく、現在から過去の数時刻分のスキャンを用いて局所地図を作成し、その局所地図と地図データとの位置合わせを行う手法を提案している。この手法では、過去 K 時刻分のスキャンを張り合わせて局所地図を生成している。しかし、局所地図生成に用いるスキャンの数を予め決定しておく、その局所地図内に環境を表す形状特徴が不足している場合には、局所地図を用いても位置合わせが困難になる。

我々の研究グループでも、オクルージョンのある環境や形状特徴の乏しい環境内でも大域的自己位置推定を成功させるために、現在と過去 (二時刻分) のスキャンの点列の順序を保存して統合したスキャン (局所地図と同じ) を生成し、このスキャンと地図とを入力として CIF-based スキャンマッチング法により大域的自己位置推定を行う手法を提案した [11]。この手法により、実環境の地図データに対しても、より頑健な位置合わせが可能となった。しかし、局所地図生成に用いるスキャン数を予め限定してしまうと、それらのスキャンの中に環境の形状特徴が含まれていない可能性があるため、望ましくない。局所地図生成に用いるスキャンの数は、観測した環境に合わせて変化させることが望まれる。そこで、本研究では、自己位置推定に用いるスキャン数を環境の形状によって決定するために、地形情報を数値化する“地形複雑度”とそれを用いて局所地図を作成する“選択的統合処理”を提案し、その処理を利用した大域的自己位置推定を行う方法についても提案する。1時刻のスキャンに環境の形状特徴が十分に含まれているか否かを判断することを2クラスの識別問題として捉え、識別器によって環境の形状特

^{*1} 連続するスキャン点を結んで得られる折れ線において角度差が急激に変化する点をキーポイントとして抽出する。抽出されたキーポイントにおいて、特徴量を求める。

徴の有無を判断する．この 2 クラスの識別問題を扱うために，本研究では，Support Vector Machine (SVM) [33, 34] を用いた識別器を利用する．スキヤンの地形複雑度と形状特徴の有無を表すラベルからなる訓練データを生成して，SVM を用いた識別器を訓練する．選択的統合処理では，訓練した識別器にスキャンから時々刻々得られる地形複雑度を入力し，スキャンに含まれる形状特徴が十分か否かを判断して，不十分であれば現在のスキャンと過去のスキャンを統合して局所地図を生成する．生成された局所地図に形状特徴が十分含まれているか否かを，同じ識別器を使って判断する．生成される局所地図に形状特徴が十分に含まれていると判定されるまで，統合処理を繰り返し，形状特徴が十分に含まれた局所地図を生成する．この選択的統合処理により，局所地図作成に用いるスキャンの数を予め決定する必要がなく，また，形状特徴が十分に含まれた局所地図を用いて地図との位置合わせを頑健に行えるようになる．

レーザスキャナから得られるスキャンを用いた自己位置推定に関する研究では，スキャンと地図の位置合わせを行うスキャンマッチング法を利用して自己位置を推定する．スキャンの値をそのまま利用して大域的自己位置推定する場合には，地図内のスキャンと観測時に得られるスキャンで形状が重なる部分が多いことが望まれるが，実際の環境では，地図作成時には無かった移動物体や静止物体によって形状の重なりが少なくなってしまう場合がある．そのため，スキャンから地形の形状を表す特徴量を生成し，その特徴量を用いてスキャンマッチングを行う手法が提案されている．

1.5.3 スキャンから特徴量を直接生成する方法

Tipaldi ら [19] や，Kallasi ら [20] や，著者ら [21] はスキャンを画像に変換することなく，スキャンそのものから特徴量を直接生成する方法を提案している．

これらの方法では，まずスキャンからコーナなどの特徴点 (キーポイント) を抽出し，キーポイントを中心として周囲を小領域で区切る．そして各小領域をヒストグラムのビンとみなし，キーポイント周辺の点の分布のヒストグラムを特徴量として生成している．これらは，実時間での処理が可能で，対応するキーポイントが複数検出できれば精度のよいマッチングができる．しかし，キーポイントを利用した特徴量を用いる場合，スキャンから検出されるキーポイントの数はスキャン毎に異なる．スキャンマッチングを行うためには，複数の対応するキーポイントが必要となるため，抽出できるキーポイントの数が少ない場合，自己位置推定の精度が不安定になる．これに対して提案手法は，スキャンから環境を表す固有の特徴量を求めるという目的は同じであるが，キーポイントを抽出する必要がなく，さらにスキャン全体から 1 つの特徴量を生成する点で異なる．

1.5.4 スキャンを画像化して特徴量を生成する方法

Li ら [16] は、スキャンから特徴量を生成する方法として画像特徴量を利用することを提案している。この方法では、スキャンを画像情報へ変換する。そうして得られた画像から Kanade-Tomasi コーナ検出器を用いてキーポイントを検出し、各キーポイントにおいて SIFT 特徴量 [35] を計算する。地図はこの特徴量を用いて予め構成しておき、生成した特徴量と地図の特徴量のマッチングを行うことで、対応箇所を求め、自己位置推定する。さらに、彼らの研究グループでは、特徴量マッチングの誤対応を軽減するために、対応点の候補を抑制する方法も提案している [17]。

阪東ら [18] は、2次元レーザスキャンの空間周波数解析に基づく大域的自己位置推定法を提案している。この研究例では、2次元レーザスキャナから得られるスキャンを一旦、2次元画像情報へ変換し、その画像に対して2次元FFTにより周波数情報へ変換したものを、局所的な環境の構造を表す特徴量としている。また、この研究例の地図データは、ある間隔毎に取得されるスキャンから特徴量を生成し、その特徴量とそれを観測した時の自己位置の組を記録したものとなっている。自己位置推定する際には、観測されたスキャンから生成した特徴量と地図データ内の特徴量との照合を行い、最も一致する地図データ内のスキャンを発見し、そのスキャンと観測されたスキャンとの間の相対的な位置姿勢を求めることで自己位置推定を行っている。

しかし、スキャンを画像へ変換して特徴量を生成する方法では、画像へ変換する際の量子化間隔によって、推定される位置姿勢の精度の良し悪しが左右される。また、スキャン内に人の足などの移動物体が写りこむことにより、その時刻のスキャンから生成される特徴量の値が、地図内の同じ場所に対応する特徴量と異なってしまい、自己位置推定に失敗する場合がある。

提案手法は、前者二つの方法 [16, 18] と同じく、スキャンから自動的に特徴量を生成するという点では同じアプローチだが、静的な環境だけでなく動的な環境であっても適用可能である点が異なる。

1.5.5 環境地図圧縮法

従来から自己位置推定に用いられてきた地図として占有格子地図 (occupancy grid map) [14, 15] がある。占有格子地図では、ロボットの走行する環境をある間隔の格子に区切り、各格子に障害物が存在する確率が入力されている。ロボットの走行する環境が広くなれば、その大きさに応じて格子数を増やさなければならず、各格子に保持する値を記憶しておくためのメモリ量も増加させなければならない。

友納 [36] は占有格子地図に観測度数を導入し、地図作成に有用でないスキャンを選択して削除することで保持するデータ量を削減する方法を提案している。この方法では、ロボットの移動中に蓄積されるスキャンのうち、地図作成に有用な時間的に新しいセンサデータが残り、古いスキャンが削除されるよ

うになっている。しかし、新しく観測されたスキャンを優先的に残す処理を行っているため、新しく観測されたスキャンに移動物体があった場合、それも優先的に残してしまい、移動物体を含んだ状態で地図が生成されてしまう可能性がある。

提案手法は、圧縮した特徴量を地図として用いることで、ロボットが走行する環境が大きくなっても保持するデータ量は劇的に増加しないという特長を持っている。

1.6 本論文の構成

本論文は全 4 章で構成されている．第 1 章では本研究の背景と目的，それに対するアプローチについて述べた．第 2 章では，スキャン内に地形の形状特徴があまりない状態において自己位置推定を行う方法について述べる．第 3 章では不変的な特徴生成手法について述べる．第 4 章では本論文の結論を述べる．

2 選択的統合処理に基づく CIF ベース大域的自己位置推定

提案手法は、CIF 特徴量 [31] を使用することを前提としている。入力として、2次元レーザスキャンを用いて、ある時刻 t において取得したスキャン $^{raw}\mathbf{P}^t = \{^{raw}p_1^t, \dots, ^{raw}p_D^t\}$ 、環境地図 $^{map}\mathbf{P}$ 、環境地図内において計算したキーポイントの集合 $^{map}\mathbf{K}$ 、時刻 $t-1$ までに取得したスキャンとキーポイントの組の集合 \mathbf{V}^{t-1} を用いる。出力は $^{map}\mathbf{P}$ に対する $^{raw}\mathbf{P}^t$ の位置姿勢 $^{global}\mathbf{q} = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{\theta})$ である。ここで、 D は 2次元レーザスキャンによって取得したスキャンを構成する点 $^{raw}p_i^t$ (スキャン点と呼ぶ) の総数である。 \mathbf{V}^{t-1} は時刻 $t-1$ までに観測されたスキャンと各スキャンにおけるキーポイントの集合の組 $(\mathbf{P}^1, \mathbf{K}^1) \sim (\mathbf{P}^{t-1}, \mathbf{K}^{t-1})$ を保持している LILO 型のデータ構造である。

Fig. 4 は、提案手法の処理手順を表している。この図に示しているように提案手法は ① 前処理 + 地形複雑度の算出、② 選択的統合処理、③ 位置姿勢推定、という 3 つのステップからなる。以下では、これら 3 つのステップについて説明する。

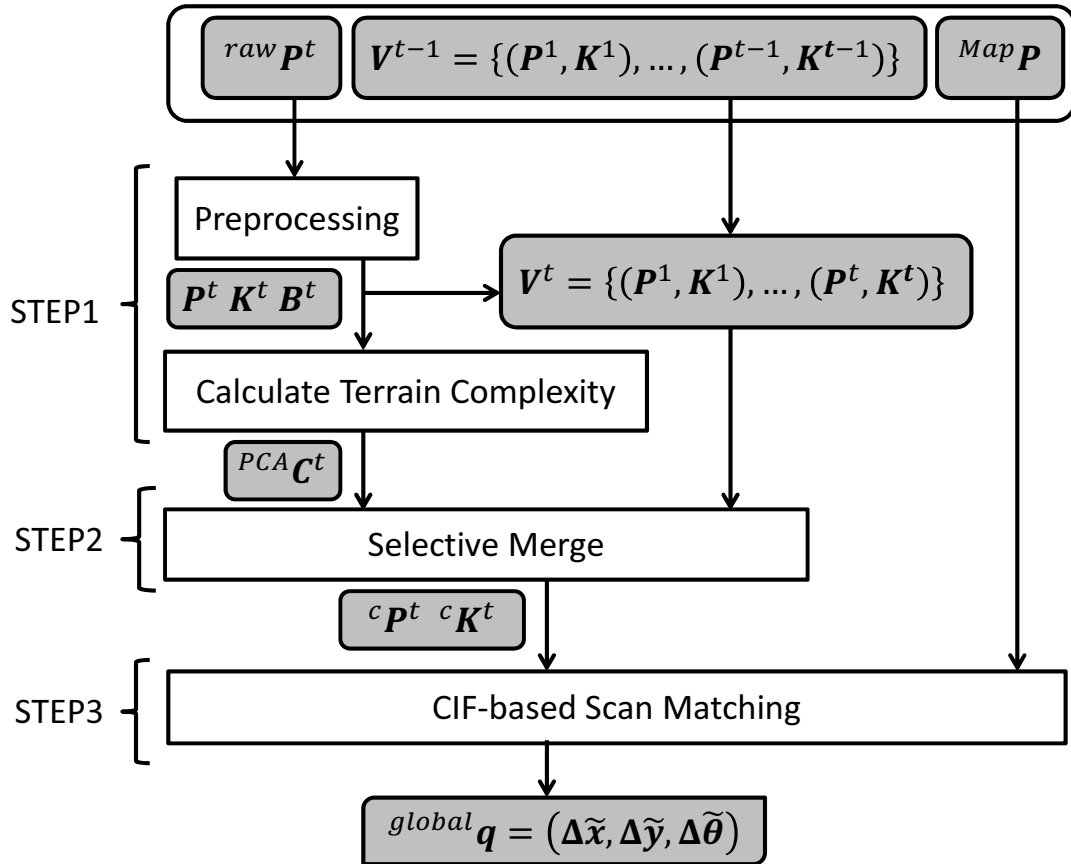


Fig. 4 Overview of our method

2.1 前処理と地形複雑度の算出

ある地点を近距離から観測したスキャンデータと遠距離から観測したスキャンデータとではその地点の近傍のスキャンデータの密度が異なっているため、そのまま CIF 特徴量の計算を行うと、計算された特徴量にずれが生じる。この影響を低減するために前処理では、2 次元レーザスキャンで取得したスキャン $raw \mathbf{P}^t$ と集合 \mathbf{V}^{t-1} を入力して、スキャン $raw \mathbf{P}^t$ を整形し、整形後のスキャンからキーポイント・不連続点を抽出して、整形後のスキャンとキーポイントの組を集合 \mathbf{V}^{t-1} に追記して、整形後のスキャン、キーポイント・不連続点の集合、集合 \mathbf{V}^t を出力する。

整形処理では、スキャン $raw \mathbf{P}^t$ を構成するスキャン点の間隔を均等にする処理を行う。これにより、点間の距離が一定になるため、後に計算する CIF 特徴量の一意性が増す。ここで、整形後のスキャン $\mathbf{P}^t = \{p_1^t, \dots, p_N^t\}$ とする。ここで、 $p_i^t = (x_i^t, y_i^t)$ である。

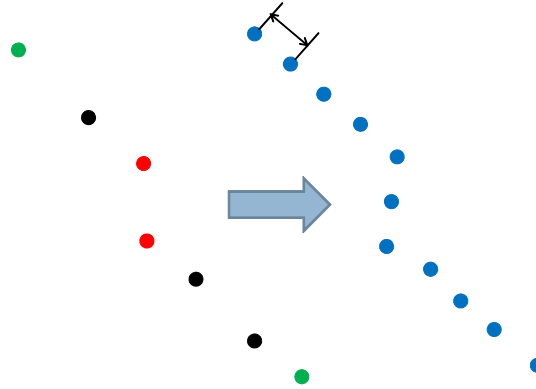


Fig. 5 Reforming gaps between scan data points

整形後のスキャン \mathbf{P}^t から、不連続点^{*2}を抽出し、抽出された不連続点をまとめて、集合を \mathbf{B}^t として記録する。不連続点は文献 [37] で提案されている手法を利用して抽出する。そして、スキャン \mathbf{P}^t からキーポイントを抽出し、抽出されたキーポイントをまとめて集合 \mathbf{K}^t として記録して、整形後のスキャンとキーポイントの組 $(\mathbf{P}^t, \mathbf{K}^t)$ を集合 \mathbf{V}^{t-1} に追記する。

次に、地形複雑度の算出方法について述べる。

地形複雑度は、整形後のスキャン \mathbf{P}^t 、キーポイントの集合 \mathbf{K}^t 、不連続点の集合 \mathbf{B}^t を用いて算出する。

形状が同じでスケールの異なる 2 つのスキャンの地形複雑度が同じ値になるように (例えば “幅 3[m] のまっすぐな廊下” と “幅 1[m] のまっすぐな廊下” の地形複雑度が同じ値になるように)、スキャン \mathbf{P}^t を

^{*2} 本論文では、連続する 2 点間の距離が 0.5m 以上離れた場合、その 2 点を不連続点とした。不連続点では、一意性のある特徴がとれないため、キーポイントとして抽出しない。

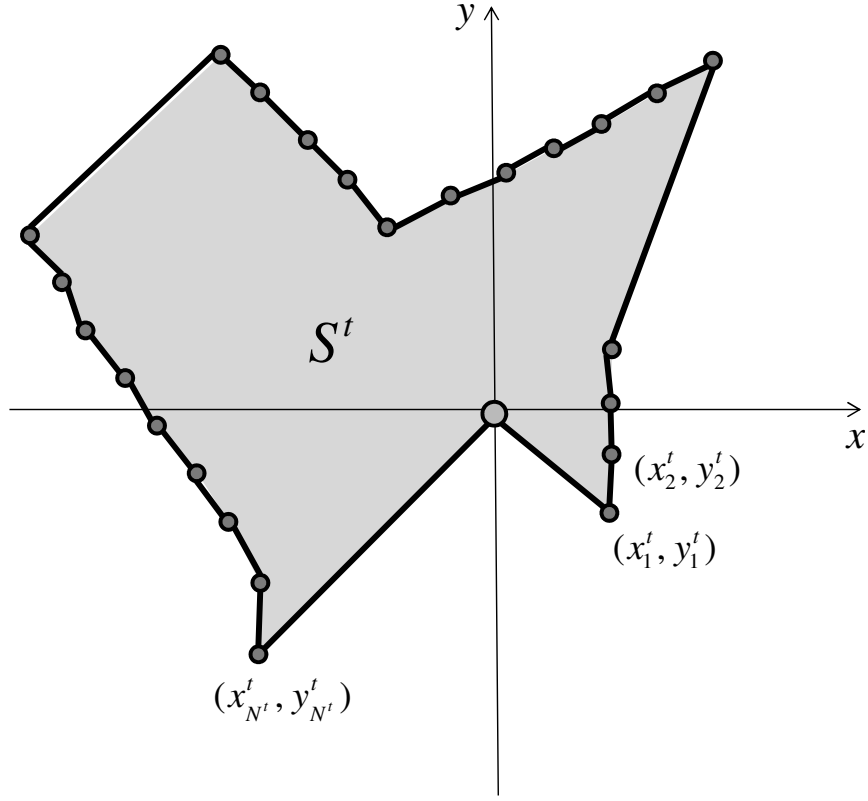


Fig. 6 How to define the area S^t

構成するスキャン点を結んで得られる多角形の面積（スキャンの面積と呼ぶ）を計算し、その面積が一定の値 cS になるようにスキャン点の座標を変換する． Fig.6 は、スキャン $\mathbf{P}^t = \{(x_1^t, y_1^t), \dots, (x_{N'}^t, y_{N'}^t)\}$ と面積 S^t の関係を表している．

スキャン $\mathbf{P}^t = \{(x_1^t, y_1^t), \dots, (x_{N'}^t, y_{N'}^t)\}$ の面積 S^t は、次式のように計算する．

$$S^t = \frac{1}{2} \left| \sum_{i=1}^{N+1} (x_i^t y_{i+1}^t - x_{i+1}^t y_i^t) \right| \quad (1)$$

ここで、ロボットの位置を表す座標として $(x_{N+1}^t, y_{N+1}^t) = (0, 0)$ と $(x_{N+2}^t, y_{N+2}^t) = (x_1^t, y_1^t)$ を、この計算の都合上スキャン点として追加して、面積の計算に用いる．こうして求められたスキャンの面積が一定の値 cS になるように、次式に従ってスキャン点の座標を変換し、変換後のスキャン ${}^{trans}\mathbf{P}^t$ を求める．

$${}^{trans}\mathbf{P}^t = \left\{ \left(\sqrt{\frac{{}^cS}{S}} x_1^t, \sqrt{\frac{{}^cS}{S}} y_1^t \right), \dots, \left(\sqrt{\frac{{}^cS}{S}} x_{N'}^t, \sqrt{\frac{{}^cS}{S}} y_{N'}^t \right) \right\} \quad (2)$$

この変換後のスキャン ${}^{trans}\mathbf{P}^t$ を用いてスキャン ${}^{raw}\mathbf{P}^t$ の地形複雑度 ${}^{raw}\mathbf{C}^t$ を計算する．地形複雑度を 9 次元ベクトル ${}^{raw}\mathbf{C}^t = ({}^{raw}\mathbf{C}_1^t, {}^{raw}\mathbf{C}_2^t, \dots, {}^{raw}\mathbf{C}_9^t)$ で表すものとし、スキャン ${}^{trans}\mathbf{P}^t$ を文献 [37] に従ってセグメンテーションした後、各次元の値は次のように決める．

$raw\mathbf{C}_1^t$ スキャン $trans\mathbf{P}^t$ の総点数

$raw\mathbf{C}_2^t$ キーポイント \mathbf{K}^t の総点数

$raw\mathbf{C}_3^t$ 不連続点 \mathbf{B}^t の総点数

$raw\mathbf{C}_4^t$ スキャン $trans\mathbf{P}^t$ 内のセグメントの総数

$raw\mathbf{C}_5^t$ セグメントの長さの平均値

$raw\mathbf{C}_6^t$ セグメントの長さの分散

$raw\mathbf{C}_7^t$ セグメント間の距離の平均値

$raw\mathbf{C}_8^t$ セグメント間の距離の分散

$raw\mathbf{C}_9^t$ スキャン $trans\mathbf{P}^t$ の周囲長

提案手法は CIF 特徴量を用いた特徴量マッチングによって自己位置推定を行う。そこで、CIF 特徴量を算出するうえで重要となる 9 個の値を用いて地形複雑度を構成する。 $raw\mathbf{C}_1^t$ はスキャン $trans\mathbf{P}^t$ の総点数を用いる。この総点数は、CIF 特徴量算出の前処理としてスキャン点列を整形しているため、点間距離がほぼ一定になっており、スキャンの面積を反映した値となっている。 $raw\mathbf{C}_2^t$ はキーポイント \mathbf{K}^t の総数を用いる。キーポイントは CIF 特徴量を算出する点を表し、キーポイントが多いほど、地図とのマッチング候補が多いことを意味する。 $raw\mathbf{C}_3^t$ は不連続点 \mathbf{B}^t の総数を用いる。CIF 特徴量はキーポイント周辺の点の分布によって計算されるが、キーポイント周辺に不連続点がある場合は点の分布に抜けが発生し、計算された CIF 特徴量の一意性が保証されない。 $raw\mathbf{C}_4^t$ はスキャン $trans\mathbf{P}^t$ 内のセグメントの総数を用いる。スキャンからセグメントを抽出することで、 $raw\mathbf{C}_5^t$ はセグメントの長さの平均値を表し $raw\mathbf{C}_6^t$ はセグメントの長さの分散を表す。これらを算出することで、スキャン内に点がどの程度まとまって分布しているかが分かる。 $raw\mathbf{C}_7^t$ はセグメント間の距離の平均値を表し $raw\mathbf{C}_8^t$ はセグメント間の距離の分散を表す。これらを算出することで、スキャン内に点がどの程度散らばって分布しているかが分かる。 $raw\mathbf{C}_9^t$ はスキャン $trans\mathbf{P}^t$ の周囲長で、スキャン内の各点を結んで得られる多角形の周囲長を計算する。面積を一定にした後の周囲長を算出することで、スキャン形状の凹凸具合が分かる。このようにこれら 9 種類^{*3}の値を用いて、スキャンの持つ様々な特性を表すことを試みている。

算出した $raw\mathbf{C}^t$ に対して主成分分析を行い、次元圧縮した $PCA\mathbf{C}^t$ を選択的統合処理で用いる地形複雑度とする。

本論文では t は累積寄与率が 90% を超える最小の次元数とした。

^{*3} ただし、この 9 種類が最適である保証はなく、冗長な表現になっていると考えている。そこで、次のように主成分分析を行っている。

2.2 選択的統合処理

選択的統合処理では、時刻 t までのスキャンとそのキーポイントの集合 \mathbf{V}^t と地形複雑度 $^{PCA}\mathbf{C}^t$ に基づいて、現在の時刻 t のスキャンに環境の形状特徴が十分に含まれているか否かを判断して、不十分であれば現在のスキャンと過去のスキャンを統合して局所地図を生成し、さらに、生成された局所地図に形状特徴が十分に含まれているか否かを判断して、生成される局所地図に形状特徴が十分に含まれていると判定されるまで、判断・統合処理を繰り返し、形状特徴が十分に含まれた局所地図を生成する。このとき、過去のスキャンが十分でない場合は、その時点で生成されている局所地図を出力とする（例えば $t = 1$ では局所地図は生成しない）。Algorithm 1 は、選択的統合処理の詳細な処理手順を表している。

Algorithm 1 SelectiveMerge($\mathbf{V}^t, ^{PCA}\mathbf{C}^t$)

Require: $\mathbf{V}^t = \{(\mathbf{P}^1, \mathbf{K}^1), \dots, (\mathbf{P}^t, \mathbf{K}^t)\}$ /*時刻 t までのスキャンとそのキーポイントの集合*/ ,
 $^{PCA}\mathbf{C}^t$ /*時刻 t のスキャンの地形複雑度*/

Ensure: $^c\mathbf{P}^t, ^c\mathbf{K}^t$ /*形状特徴の十分なスキャンとそのキーポイントの集合*/

- 1: $^c\mathbf{P}^t, ^c\mathbf{K}^t \leftarrow \text{popScan}(\mathbf{V}^t)$
- 2: **for** $s = t - 1$ **to** 1 **do**
- 3: /*スキャン形状の判断*/
- 4: **if** ShapeJudge($^{PCA}\mathbf{C}^t$) = Yes **then**
- 5: **return** $^c\mathbf{P}^t, ^c\mathbf{K}^t$
- 6: **else**
- 7: $\mathbf{P}^s, \mathbf{K}^s \leftarrow \text{popScan}(\mathbf{V}^s)$
- 8: $^{local}\mathbf{q} \leftarrow \text{CIF-basedScanMatching}(^c\mathbf{P}^t, ^c\mathbf{K}^t, \mathbf{P}^s, \mathbf{K}^s)$
- 9: $\mathbf{P}^s \leftarrow \text{Transform}(\mathbf{P}^s, ^{local}\mathbf{q})$
- 10: $^{merge}\mathbf{P}^t \leftarrow \text{OrderPreservingMerge}(^c\mathbf{P}^t, \mathbf{P}^s)$
- 11: /*前処理*/
- 12: $^c\mathbf{P}^t, ^c\mathbf{B}^t, ^c\mathbf{K}^t \leftarrow \text{Preprocessing}(^{merge}\mathbf{P}^t)$
- 13: /*地形複雑度の計算*/
- 14: $^{PCA}\mathbf{C}^t \leftarrow \text{CalculateTerrainComplexity}(^c\mathbf{P}^t, ^c\mathbf{K}^t, ^c\mathbf{B}^t)$
- 15: **end if**
- 16: **end for**
- 17: **return** $^c\mathbf{P}^t, ^c\mathbf{K}^t$

以下では、この Algorithm 1 内の関数について説明する。

関数 popScan() は、LILO 型データ構造から、pop して最後の要素を 1 つ取り出す関数である。従って、同じ LILO 型のデータ構造がこの関数入力となっている場合、pop される毎に一時刻前のデータが

取り出されることになる．この関数によって， \mathbf{V}^t から取り出されたスキャンとキーポイントの集合の組は， ${}^c\mathbf{P}^t$ ， ${}^c\mathbf{K}^t$ に保存しておく．また，この関数の入力が，時刻 s までのスキャンとそのキーポイントの集合 \mathbf{V}^s （ただし， t は現在時刻を表し，時刻 s には， $t-1$ から順に過去に遡って代入される）の場合，取り出されるデータを，過去のスキャンとキーポイントの集合の組 \mathbf{P}^s ， \mathbf{K}^s に保存しておく．

関数 ShapeJudge() は，地形複雑度 ${}^{PCA}\mathbf{C}^t$ を入力として， ${}^{PCA}\mathbf{C}^t$ のもとのスキャン ${}^c\mathbf{P}^t$ に形状特徴が十分含まれている（Yes）か，含まれていない（No）かを判断する関数である．この判断を下すために，本研究では，SVM に基づく識別器を用いる^{*4}．関数 ShapeJudge() の識別器の実体は，学習された SVM に基づく識別器である．この SVM に基づく識別器の学習は，次のようにして行う．まず，通常の CIF-based スキャンマッチング法 [11] を用いて，大域的自己位置推定する環境でスキャンマッチングに成功するスキャン，失敗するスキャンを多数収集し，それぞれのスキャンに対して，スキャンマッチングに成功する・失敗するラベルを付与して，先述した地形複雑度を求める．こうして，スキャンマッチングに成功する・失敗するラベルと，それらに対応した地形複雑度の組合せのデータを作成する．その後，このデータにおいて，スキャンマッチングに成功するラベルを形状特徴が十分であるラベルに替え，スキャンマッチングに失敗するラベルを形状特徴が不十分であるラベルに替えて，これを SVM に基づく識別器の訓練データとして用いて，識別器を学習する．関数 ShapeJudge() の識別器により，スキャン ${}^c\mathbf{P}^t$ の形状特徴が十分と判断された場合は，関数 SelectiveMerge() は，選択的統合処理の結果として，スキャン ${}^c\mathbf{P}^t$ とキーポイントの集合 ${}^c\mathbf{K}^t$ の組を出力する．関数 ShapeJudge() の識別器により，スキャン ${}^c\mathbf{P}^t$ の形状特徴が不十分と判断された場合は，次のような処理を順次実行していく．まず，時刻 s までのスキャンとそのキーポイントの集合 \mathbf{V}^s （ただし， t は現在時刻を表し，時刻 s は， $s < t$ である．）から，過去のスキャンとキーポイントの集合の組 \mathbf{P}^s ， \mathbf{K}^s を取り出す．その後，関数 CIF-basedScanMatching() を用いて，現在のスキャン ${}^c\mathbf{P}^t$ と過去のスキャン \mathbf{P}^s の間の相対姿勢 ${}^{local}\mathbf{q} = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{\theta})$ を求める．関数 CIF-basedScanMatching() は我々が提案している CIF-based スキャンマッチング法で．詳細は 2.3 節で述べる．

関数 Transform() は，関数 CIF-basedScanMatching() により求められた相対位置姿勢 ${}^{local}\mathbf{q}$ を用いて，スキャン \mathbf{P}^s のデータをスキャン ${}^c\mathbf{P}^t$ のデータの座標系におけるスキャン \mathbf{P}^s に変換する．

関数 OrderPreservingMerge() は， ${}^c\mathbf{P}^t$ と \mathbf{P}^s を入力して，これら 2 つのスキャンを点列の順序を保存して統合したスキャン ${}^{merge}\mathbf{P}^t$ を出力する．この統合したスキャン ${}^{merge}\mathbf{P}^t$ が局所地図になる．

^{*4} SVM は，教師あり学習を行う学習器のひとつで，2 クラスの分類を行う．SVM では，複雑な分布をしている学習データであっても，カーネル関数を用いて高次元の空間に写像することで，線形モデルとして扱うことができる．これを用いることで，地形複雑度が複雑な分布をしていても 2 クラスの識別が可能になる．

Algorithm 2 OrderPreservingMerge(${}^cP^t, P^s$)

Require: ${}^cP^t = \{{}^cp_1^t, \dots, {}^cp_{N^t}^t\}$, $P^s = \{p_1^s, \dots, p_{N^s}^s\}$ **Ensure:** ${}^{merge}P^t$

```
1:  $L \leftarrow \text{RangeSearch}({}^cP^t, P^s)$ 
2: if ( $L(1) \neq \emptyset$  and  $L(1) > 1$ ) then
3:   /*  ${}^cp_1^t$  以前にある  $P^s$  内の点を追加 */
4:   for  $j = 1$  to  $L(1) - 1$  do
5:      ${}^{merge}P^t \leftarrow \text{addPoint}(p_j^s)$ 
6:   end for
7: end if
8:  ${}^{merge}P^t \leftarrow \text{addPoint}({}^cp_1^t)$ 
9: for  $i = 1$  to  ${}^cN^t - 1$  do
10:  if ( $L(i) \neq \emptyset$  and  $L(i+1) \neq \emptyset$  and  $(L(i+1) - L(i)) > 1$ ) then
11:    /*  ${}^cp_i^t$  と  ${}^cp_{i+1}^t$  の間にある  $P^s$  内の点を追加 */
12:    for  $j = L(i) + 1$  to  $L(i+1) - 1$  do
13:       ${}^{merge}P^t \leftarrow \text{addPoint}(p_j^s)$ 
14:    end for
15:  end if
16:  /* 上記以外の場合は,  ${}^cp_{i+1}^t$  を追加 */
17:   ${}^{merge}P^t \leftarrow \text{addPoint}({}^cp_{i+1}^t)$ 
18: end for
19: if ( $L({}^cN^t) \neq \emptyset$  and  $L({}^cN^t) < N^s$ ) then
20:  /*  ${}^cp_{N^t}$  以降にある  $P^s$  内の点を追加 */
21:  for  $L({}^cN^t) + 1$  to  $N^s$  do
22:     ${}^{merge}P^t \leftarrow \text{addPoint}(p_j^s)$ 
23:  end for
24: end if
```

Algorithm 2 は、関数 OrderPreservingMerge() の詳しい処理手順を示している。また、Fig. 7 は、2 つのスキャンを統合したスキャン (局所地図) が生成される過程を表している。

関数 OrderPreservingMerge() の中で、 L は、 ${}^cP^t$ 内の点と同一の位置に存在すると見なせる P^s 内の点の番号を保存するためのリストである。初期状態で、 ${}^{merge}P^t = \{\emptyset\}$ かつ $L = \{\emptyset, \dots, \emptyset\}$ である。

関数 RangeSearch() は、 ${}^cP^t$ 内の点 ${}^cp_i^t$ を中心として距離 ϵ 内 (Fig. 7 内で丸で囲んだ部分) に存在する P^s 内の点の番号を、 ${}^cP^t$ 内の全ての点について求めて、その番号をリスト L に保存する。距離 ϵ 内に存在する点がいなければ、その点の番号は \emptyset となる。

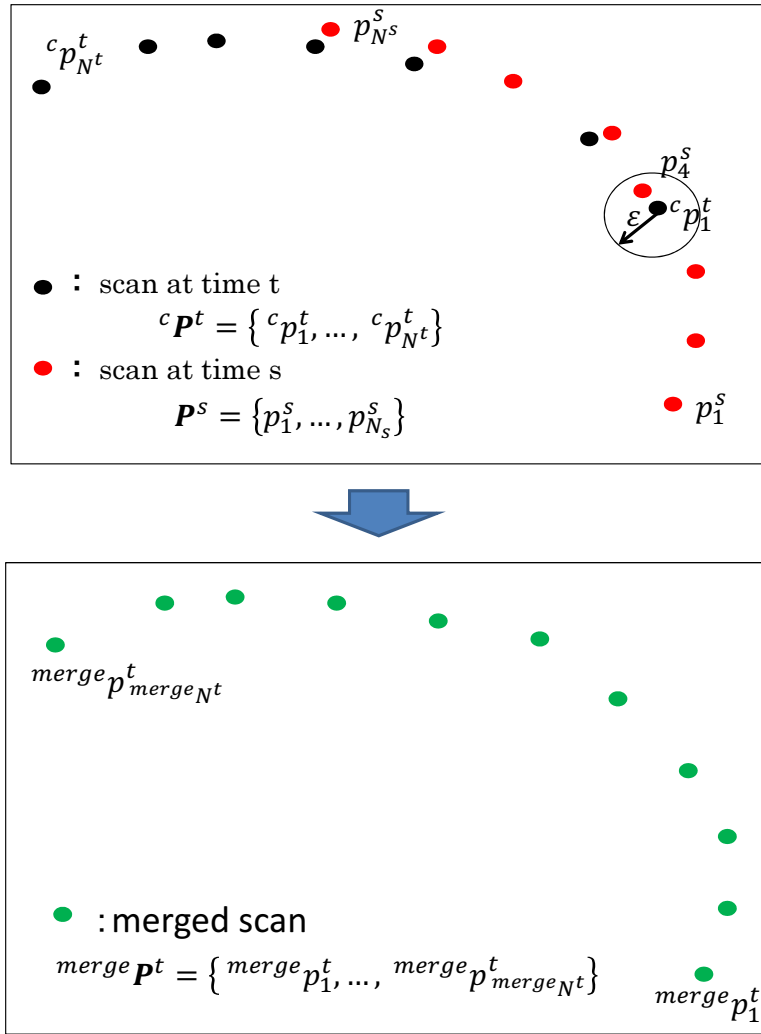


Fig. 7 How to perform order-preserving merge

まず, $c\mathbf{P}^t$ 以前にある \mathbf{P}^s 内の点を $\text{merge}\mathbf{P}^t$ に追加する. 次に $c\mathbf{P}^t$ を $\text{merge}\mathbf{P}^t$ に追加していく. このとき, $c\mathbf{P}^t$, $c\mathbf{P}^{t+1}$ 間に \mathbf{P}^s 内の点があれば順に $\text{merge}\mathbf{P}^t$ に追加する. その後 $c\mathbf{P}^t$ 以前にある \mathbf{P}^s 内の点を $\text{merge}\mathbf{P}^t$ に追加し $\text{merge}\mathbf{P}^t$ を得る.

そして最後に, スキャン $\text{merge}\mathbf{P}^t$ を新たな入力として, 前処理, 地形複雑度の算出, スキャンの形状判断を行う.

最終的に, このような統合処理をスキャンの形状特徴が十分になるまで繰り返し, 形状特徴が十分なスキャン (局所地図) とそのスキャンのキーポイントの集合の組 $c\mathbf{P}^t$, $c\mathbf{K}^t$ を生成する.

2.3 CIF-based スキャンマッチング

Algorithm 3 は、これまで我々が提案してきた CIF-based スキャンマッチング法（関数 CIF-basedScanMatching()）の詳細な処理手順を表している．関数 CIF-basedScanMatching() は、2 つのスキャン $^{ref}\mathbf{P}$, $^{cur}\mathbf{P}$ と各スキャンにおけるキーポイントの集合 $^{ref}\mathbf{K}$, $^{cur}\mathbf{K}$ を入力として与えると、 $^{ref}\mathbf{P}$ に対する $^{cur}\mathbf{P}$ の相対的な位置姿勢 $\mathbf{q} = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{\theta})$ を求める．このとき、相対的な位置姿勢の初期値を与える必要はない．

Algorithm 3 CIF-basedScanMatching($^{ref}\mathbf{P}$, $^{ref}\mathbf{K}$,
 $^{cur}\mathbf{P}$, $^{cur}\mathbf{K}$)

Require: $^{ref}\mathbf{P}$, $^{ref}\mathbf{K}$, $^{cur}\mathbf{P}$, $^{cur}\mathbf{K}$

Ensure: $\mathbf{q} = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{\theta})$

```

1: /*Step1 : 特徴量の算出*/
2:  $^{ref}\mathbf{H} \leftarrow \text{CalculateCIFdescriptors}(^{ref}\mathbf{P}, ^{ref}\mathbf{K})$ 
3:  $^{cur}\mathbf{H} \leftarrow \text{CalculateCIFdescriptors}(^{cur}\mathbf{P}, ^{cur}\mathbf{K})$ 
4: /*Step2 : 対応点探索*/
5:  $\mathbf{T} \leftarrow \text{FindCorrespondingPoints}(^{ref}\mathbf{K}, ^{cur}\mathbf{K}, ^{ref}\mathbf{H}, ^{cur}\mathbf{H})$ 
6: /*Step3 : 粗な位置姿勢推定*/
7:  $^{rough}\mathbf{q} \leftarrow \text{RoughMatching}(\mathbf{T})$ 
8: /*Step4 : 詳細マッチング*/
9:  $\mathbf{q} \leftarrow \text{ICP}(^{rough}\mathbf{q}, ^{ref}\mathbf{P}, ^{cur}\mathbf{P}, \mathbf{T})$ 
10: return  $\mathbf{q}$ 

```

関数 CIF-basedScanMatching() によって大域的自己位置推定を実行する際には、 $^{ref}\mathbf{P}$ として $^{map}\mathbf{P}$ （大域的地図に相当する）、 $^{cur}\mathbf{P}$ として $^c\mathbf{P}^t$ を用いる．選択的統合処理の中で関数 CIF-basedScanMatching() を使用する際には、 $^{ref}\mathbf{P}$ として $^c\mathbf{P}^t$ （局所地図に相当する）、 $^{cur}\mathbf{P}$ として \mathbf{P}^s を用いる．

CIF-based スキャンマッチング法は 3 つのステップ ① 特徴量の算出、② 対応点探索、③ スキャンマッチングからなる．

ステップ ① では、入力された各スキャン $^{ref}\mathbf{P}$, $^{cur}\mathbf{P}$ の CIF 特徴量を算出する．ステップ ① に相当する関数 CalculateCIFdescriptors() は、スキャンとキーポイントの集合を入力として与えると、各キーポイントにおける CIF 特徴量を出力する．Algorithm 3 中の変数 $^{ref}\mathbf{H}$, $^{cur}\mathbf{H}$ は各スキャン $^{ref}\mathbf{P}$, $^{cur}\mathbf{P}$ の CIF 特徴量の集合を表している．

ステップ ② に相当する関数 FindCorrespondingPoints() は、2 つのスキャンにおけるキーポイントの集合と、キーポイントにおける CIF 特徴量の集合を入力として与えると、それぞれのスキャンにおける 3 個のキーポイントの組合せを全て求め、2 つのスキャンにおいて CIF 特徴量が最も一致するキーポイントの組合せを、2 つの対応するキーポイントの組合せ $T = \{(^{ref}P_{\alpha 1}, ^{ref}P_{\alpha 2}, ^{ref}P_{\alpha 3}), (^{cur}P_{\beta 1}, ^{cur}P_{\beta 2}, ^{cur}P_{\beta 3})\}$ を出力する。ここで、 αi と $\beta i (i = 1 \sim 3)$ が対応している 2 つのキーポイントの番号を表している。

ステップ ③ に相当する関数は、RoughMatching() と ICP() で、RoughMatching() は、2 つの対応するキーポイントの組合せ T を入力として与えると、 ^{ref}P に対する ^{cur}P の相対的な位置姿勢 $^{rough}q$ を出力する。ここで求められる位置姿勢 $^{rough}q$ は、2 つの対応するキーポイントの組合せのみのデータしか用いていないため、2 つのスキャンを構成するスキャン点同士の多くの部分を位置合わせ出来るような解となっていないので、相対的な位置姿勢の近似解となる。そこで、求めた近似解 $^{rough}q$ とスキャン ^{ref}P , ^{cur}P , さらに 2 つの対応するキーポイントの組合せ T を用いて、ICP アルゴリズム [38, 4] によって、 ^{ref}P に対する ^{cur}P の相対的な位置姿勢の詳細解 q を関数 ICP() によって求める。ICP() は ICP アルゴリズム [38, 4] を実行する関数である。こうして求められた位置姿勢の詳細解 q が、関数 CIF-basedScanMatching() の出力となる。

2.3.1 特徴量の算出

CIF 特徴量の算出手順を説明する。レーザーレンジファインダから得られたスキャンの連続する 2 点を結んだ時に得られる折れ線において、各線分の傾きが急激に変化する点をキーポイントとして検出する。図 8(a) ように、スキャン P^t が得られた時、そのスキャン点間 $p_i, p_{i+\alpha}$ を結ぶ勾配方向 η_i を算出する。すべてのスキャン点における勾配方向 η_i は、レーザーレンジファインダはスキャン点が順番に沿って得られるため、点を並び変える必要がなく容易に求めることができる。

$$\eta_i = \arctan \left(\frac{y_{i+\alpha} - y_i}{x_{i+\alpha} - x_i} \right)$$

求まった勾配方向 η_i から点間の線分の勾配方向の変化率を求める。

$$\vartheta_i = \eta_{i+1} - \eta_i$$

求まった勾配方向の変化率がある閾値 $thresh$ を超えた点をキーポイント*5とする。また、キーポイントの抽出は、あくまで、スキャンを構成する点群の中から、次に計算する CIF 特徴量を計算する点を選

*5 なお、実際にキーポイントを検出する際には、 p_i から 100[mm] 以上離れている近傍の点を $p_{i+\alpha}$ として線分の勾配方向を計算している。

択するために行っているだけであるので，抽出の際に設定する閾値 $thresh$ は，厳密に選定しなくても良い．

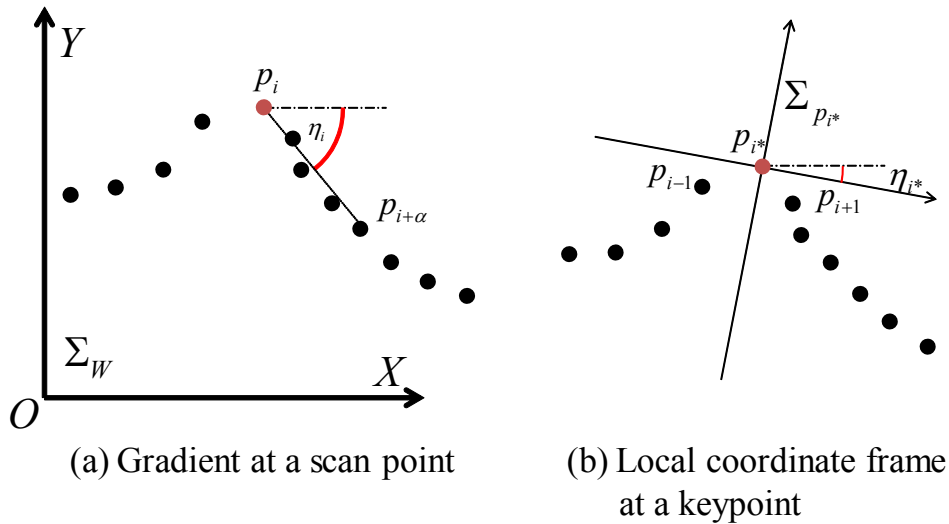


Fig. 8 Definition of a keypoint

得られたキーポイント p_{i*} を中心に，ある距離内に存在する周囲の点の配置から，CIF 特徴量を算出する．まず，図 8(b) ように，キーポイント p_{i*} を中心とした局所的な座標系 $\Sigma_{p_{i*}}$ を設定する．

キーポイントの前後 p_{i-1}, p_{i+1} に位置する点^{*6}とキーポイント p_{i*} を結んでできる 2 つの線分の勾配方向を求める．求められた 2 つの勾配方向の平均値 η_{i*} が，キーポイント p_{i*} における局所座標系の座標軸の勾配方向となる．このようにして，キーポイント p_{i*} における局所座標系 $\Sigma_{p_{i*}}$ に関して特徴量を記述するために，キーポイントにおいて記述される特徴量は，入力スキャンと参照スキャンの姿勢に関わらず，合同変換（回転・並進変換）に関して不変となる．

キーポイント p_{i*} における特徴量は，キーポイント p_{i*} と p_{i*} の近傍に存在するスキャン点同士を結んでできる線分群の局所座標系 $\Sigma_{p_{i*}}$ における勾配方向に関するヒストグラムにより記述する．このとき，2 種類の線分を生成する．1 つの線分 SP は，キーポイント p_{i*} を起点として p_{i*} の近傍に存在するスキャン点群を結んでできる線分であり，このようにして生成される線分群は，キーポイント p_{i*} 周りのスキャン点の分布を表す．図 9(a) に，線分 SP が生成される様子を示す．

もう 1 つの線分 SD は，起点となる点と n 個先のスキャン点を結び，かつ起点となる点を 1 つずつずらして生成できる線分であり，このようにして生成される線分群は，キーポイント p_{i*} 周りに存在す

^{*6} なお，実際には， p_i から 100[mm] 以上離れている近傍の点を p_{i-1}, p_{i+1} として線分の勾配方向を計算している．

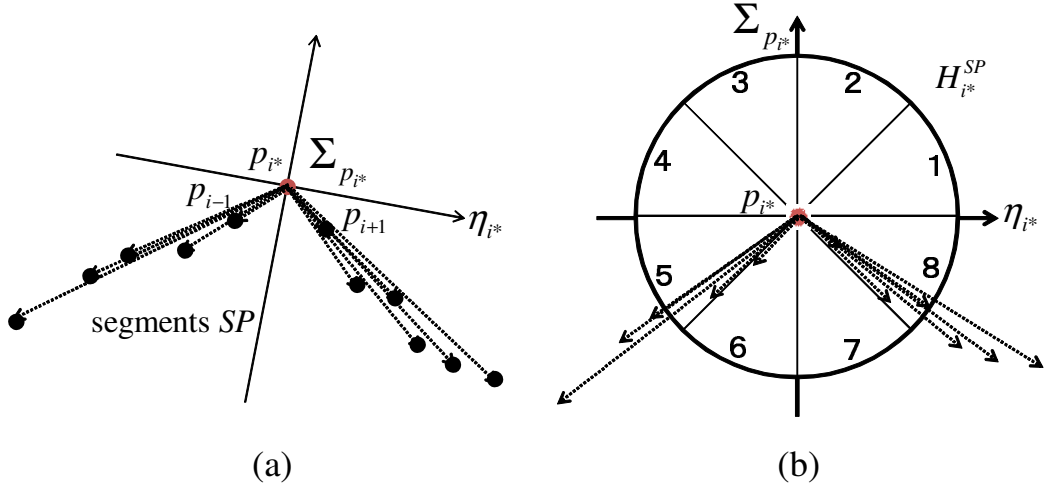


Fig. 9 Definition of segment SP and its histogram

る線分の方角の分布を表す．図 10(a) に，線分 SD が生成される様子を示す．

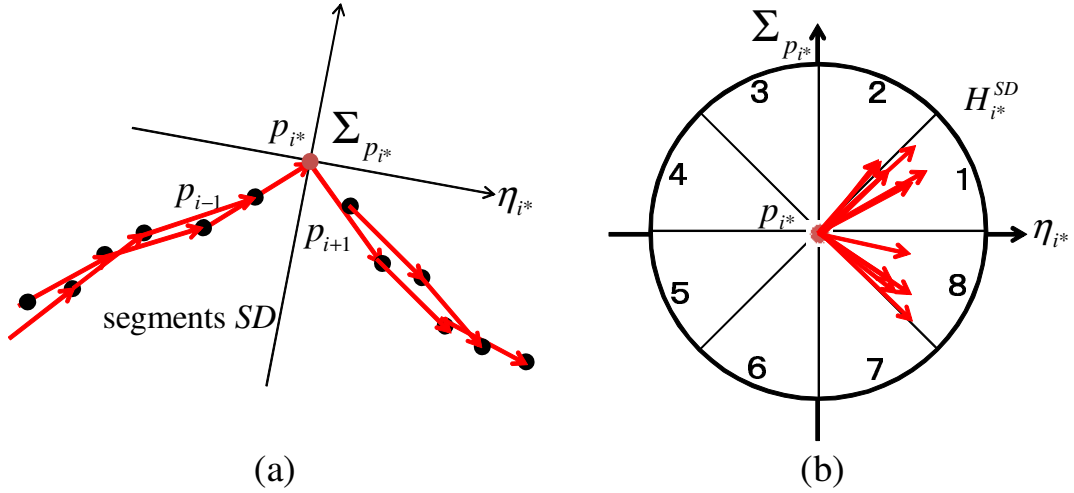


Fig. 10 Definition of segment SD and its histogram

キーポイント p_{i*} から距離 d 以内の範囲にあるスキャン点のうち，近傍にある M 個の点をリスト $neighbor(i*)$ に登録する．リスト $neighbor(i*)$ 内の点を index 順にソートしておく．リスト $neighbor(i*)$ 内のスキャン点に関して線分 SP の勾配方向 λ_i^{SP} を算出し， η_{i*} との差をとることで，局所座標系 $\Sigma_{p_{i*}}$ における線分の方角 θ_i^{SP} を求める．キーポイント p_{i*} の近傍に存在する全ての θ_i^{SP} を， $-\pi \sim \pi$ の範囲の角度を 8 方向に量子化した各ビンに分類してヒストグラム H_{i*}^{SP} を作成する．図 9(b) に，ヒストグラム H_{i*}^{SP} が生成される様子を示す．このヒストグラム H_{i*}^{SP} の各ビンの値は $H_{i*}^{SP}[k], (k = 1, 2, \dots, 8)$ と表せる．

同様にして，リスト $neighbor(i*)$ 内のスキャン点に関して線分 SD の勾配方向 λ_i^{SD} を算出し， η_{i*}

との差をとることで、局所座標系 $\Sigma_{p_{i*}}$ における線分の勾配方向 θ_i^{SD} を求める。キーポイント p_{i*} の近傍に存在する全ての θ_i^{SD} を、 $-\pi \sim \pi$ の範囲の角度を 8 方向に量子化した各ビンに分類してヒストグラム H_{i*}^{SD} を作成する。図 10(b) に、ヒストグラム H_{i*}^{SD} が生成される様子を示す。このヒストグラム H_{i*}^{SD} の各ビンの値は $H_{i*}^{SD}[k], (k = 1, 2, \dots, 8)$ と表せる。

キーポイント p_{i*} における CIF 特徴量は、キーポイント近傍の周囲 M 個から算出したヒストグラム H_{i*}^{SP} と H_{i*}^{SD} の各ビンの頻度を要素とする 16 次元のベクトル $\mathbf{H}_{i*}^M = (H_{i*}^{SP}[1], \dots, H_{i*}^{SP}[8], H_{i*}^{SD}[1], \dots, H_{i*}^{SD}[8])$ と、周囲 $2M$ 個から算出した \mathbf{H}_{i*}^{2M} を用いて合計 32 次元の特徴量 \mathbf{H}_{i*} として算出される。

$$\mathbf{H}_{i*} = \left(\mathbf{H}_{i*}^M[1], \dots, \mathbf{H}_{i*}^M[16], \frac{\mathbf{H}_{i*}^{2M}[1]}{2}, \dots, \frac{\mathbf{H}_{i*}^{2M}[16]}{2} \right) \quad (3)$$

キーポイント p_{i*} において、周囲の点 $2M$ 個から算出した \mathbf{H}_{i*}^M はヒストグラムのビン内のカウンタ数の合計が周囲の点 M 個から算出した \mathbf{H}_{i*}^M と比べて 2 倍になっているので、これを正規化するために、 \mathbf{H}_{i*}^M の各要素値に $1/2$ を乗算している。この特徴ベクトルの前半 16 次元はキーポイント近傍の狭い範囲の詳細な点の分布を表し、後半 16 次元はキーポイント周辺の広い範囲の粗な点の分布を表す。

2.3.2 対応点探索

対応点探索の処理内容は、キーポイントの幾何学的拘束に基づいて、入力スキャン内のキーポイントの集合 $^{cur}\mathbf{K}$ と参照スキャン内のキーポイントの集合 $^{ref}\mathbf{K}$ から、対応点の候補を見つけ、その各キーポイントの CIF 特徴量の合計を比較することで、対応点を探索する。

まず、入力スキャンのキーポイントの中から 3 点の組み合わせを抽出する。3 点の組み合わせを (p_s, p_t, p_u) と表す。また、キーポイント p_1, p_2 の距離を $d(p_1, p_2)$ と表す。隣接するキーポイントの特徴量は類似した値となるため、3 点の組み合わせによる一意性が向上しなくなる。そこで、3 点のキーポイント間の各距離が d_{\min} 以上 d_{\max} 以下になるように、探索するキーポイント 3 点の拘束条件を次のように定める。

$$\begin{cases} d_{\min} < d(p_s, p_t) < d_{\max} \\ d_{\min} < d(p_t, p_u) < d_{\max} \\ d_{\min} < d(p_u, p_s) < d_{\max} \end{cases} \quad (4)$$

また、入力スキャンにおいては、センサーから得られる点群はセンサー位置から遠くなればなるほど精度が落ちるため、探索するキーポイントはセンサーからの距離が r_L 以下であるとする。入力スキャンの原点を p_o とし、先の条件に加えて次のような条件を定める。

$$d(p_s, p_o) < r_L, \quad d(p_t, p_o) < r_L, \quad d(p_u, p_o) < r_L \quad (5)$$

これらの条件を満たす 3 点の組み合わせを入力スキャンのキーポイントの集合 $^{cur}\mathbf{K}$ から総当たりで探索し、 N_α 個得られたとして、それらの組合せ含むリスト ^{cur}L を生成する。

$$^{cur}L : \{(p_{s1}, p_{t1}, p_{u1}), (p_{s2}, p_{t2}, p_{u2}), \dots, (p_{sN_\alpha}, p_{tN_\alpha}, p_{uN_\alpha})\} \quad (6)$$

次に、参照スキャン内のキーポイントの集合 $^{ref}\mathbf{K}$ から、3 点の組合せを複数抽出する。

抽出する 3 点のキーポイントの組合せを (q_l, q_m, q_n) と表す。参照スキャン内のキーポイントから 3 点の組わせを抽出する際には、入力スキャン内から選択した 3 点の組合せとほぼ同じ配置にある 3 点を見つける。このように選択するために、以下の条件を設定する。この拘束条件は、3 点を結んで得られる三角形が合同であることを表している。^{*7}

$$\begin{cases} |d(p_s, p_t) - d(q_l, q_m)| < d_{th} \\ |d(p_t, p_u) - d(q_m, q_n)| < d_{th} \\ |d(p_u, p_s) - d(q_n, q_l)| < d_{th} \end{cases} \quad (7)$$

この条件を満たす 3 点の組合せが、参照スキャン内のキーポイント \mathbf{q}_{j^*} , ($j^* = 1 \sim N_{KR}$) から N_β 個得られたとして、それらの組合せ含むリスト ^{ref}L を生成する。

$$^{ref}L : \{(q_{l1}, q_{m1}, q_{n1}), (q_{l2}, q_{m2}, q_{n2}), \dots, (q_{lN_\beta}, q_{mN_\beta}, q_{nN_\beta})\} \quad (8)$$

抽出された組み合わせにおける各キーポイントの CIF 特徴量を合計し、96 次元の特徴ベクトルとする。この特徴量を比較し、最も一致するものを対応点とする。

リスト ^{cur}L 内の α 番目の組合せの 3 点における CIF 特徴量を 3 つ並べて、96 次元のベクトル \mathbf{H}_α を生成する。同様に、リスト ^{ref}L 内の β 番目の組合せの 3 点における CIF 特徴量を 3 つ並べて、96 次元のベクトル \mathbf{H}_β を生成する。これらのベクトルは、それぞれ次式のように表せる。

$$\begin{aligned} \mathbf{H}_\alpha &= (\mathbf{H}_s[1], \dots, \mathbf{H}_s[32], \mathbf{H}_t[1], \dots, \mathbf{H}_t[32], \mathbf{H}_u[1], \dots, \mathbf{H}_u[32]) \\ \mathbf{H}_\beta &= (\mathbf{H}_l[1], \dots, \mathbf{H}_l[32], \mathbf{H}_m[1], \dots, \mathbf{H}_m[32], \mathbf{H}_n[1], \dots, \mathbf{H}_n[32]) \end{aligned} \quad (9)$$

これら 2 つのベクトルの類似度を表すために、バタチャリア係数を用いる。バタチャリア係数は、ヒストグラムの一致度を求める指標で、この値が 1 に近ければ一致度が高いことになる。バタチャリア係数を利用して一致度を計算するための準備として、96 次元の特徴量ベクトルを次式に基づいて最小値 0、最大値 1 に正規化しておく。ここでは、 \mathbf{H}_α を正規化して、 $\tilde{\mathbf{H}}_\alpha$ を得る。

$$\tilde{\mathbf{H}}_\alpha[k] = \frac{\mathbf{H}_\alpha[k]}{\sum_{k=1}^{96} \mathbf{H}_\alpha[k]} \quad (10)$$

^{*7} 三角形の各辺の長さを比較し、長さの差が d_{th} より小さい場合三角形が合同になるとみなしている。本研究では $d_{th} = 0.1[\text{m}]$ とした。

このようにして正規化された、 $\tilde{\mathbf{H}}_\alpha$ と $\tilde{\mathbf{H}}_\beta$ について以下の式に従ってバチャタリア係数を計算して、入力スキンのキーポイントから抽出された 3 点の組合せ $(p_{s\alpha}, p_{t\alpha}, p_{u\alpha})$ と参照スキンのキーポイントから抽出された 3 点の組合せ $(q_{l\beta}, q_{m\beta}, q_{n\beta})$ 間の一致度 $S(\alpha, \beta)$ を求める。

$$S(\alpha, \beta) = \sum_{k=1}^{96} \sqrt{\tilde{\mathbf{H}}_\alpha[k] \tilde{\mathbf{H}}_\beta[k]} \quad (11)$$

3 点の組合せ $(p_{s\alpha}, p_{t\alpha}, p_{u\alpha})$ と $(q_{l\beta}, q_{m\beta}, q_{n\beta})$ について、 $\alpha = 1 \sim N_\alpha, \beta = 1 \sim N_\beta$ の中から、一致度が最大のものを以下の式に従って探索することによって求める。

$$(\alpha^*, \beta^*) = \arg \max_{\alpha} \left\{ \arg \max_{\beta} S(\alpha, \beta) \right\} \quad (12)$$

このようにして、一致度が最大となる対応点の組、

$$T = \{(p_{s\alpha^*}, p_{t\alpha^*}, p_{u\alpha^*}), (q_{l\beta^*}, q_{m\beta^*}, q_{n\beta^*})\} \text{ が得られる。}$$

本手法は、キーポイント間の幾何学的拘束を考慮して CIF 特徴量の次元を拡張して、CIF 特徴量の組合せで環境構造を表現することで、特徴量の一意性を向上させる。

2.3.3 スキャンマッチング

ここでは、CIF 特徴量とキーポイントの幾何学的拘束を用いたスキャンマッチングアルゴリズムを、以下のような 2 段階の処理によって構築している。

1. 粗な張り合わせ処理:

対応するキーポイント数組を用いると、入力スキンと参照スキン間の相対的な位置姿勢の変化量の閉形式解を得ることが出来る。この位置姿勢の変化量の推定の詳細については、[39, 40] を参照願いたい。入力スキンの位置姿勢と参照スキンの位置姿勢との間で位置姿勢が、変化量 $\mathbf{q} = (l_x, l_y, \delta\theta)$ だけ変化したとする。2 つのスキन्दータにおいて、対応する特徴点の組 $\mathbf{p} = (x_i, y_i)$, $\mathbf{p}' = (x'_i, y'_i)$ が定まっている場合、位置姿勢の変化量 \mathbf{q} は、以下の式で求められる

$$\begin{aligned} l_x &= \bar{x}' - (\bar{x} \cos \delta\theta - \bar{y} \sin \delta\theta) \\ l_y &= \bar{y}' - (\bar{x} \sin \delta\theta + \bar{y} \cos \delta\theta) \\ \delta\theta &= \arctan \frac{S_{xy'} - S_{yx'}}{S_{xx'} - S_{yy'}} \end{aligned} \quad (13)$$

また、ここで $\bar{x}, \bar{y}, \bar{x}', \bar{y}', S_{xx'}, S_{yy'}, S_{xy'}, S_{yx'}$ は、それぞれ次式で定義されている項目である。

$$\begin{aligned} S_{xy'} &= \sum_{i=1}^n (x_i - \bar{x})(y'_i - \bar{y}'), S_{yx'} = \sum_{i=1}^n (y_i - \bar{y})(x'_i - \bar{x}') \\ S_{xx'} &= \sum_{i=1}^n (x_i - \bar{x})(x'_i - \bar{x}'), S_{yy'} = \sum_{i=1}^n (y_i - \bar{y})(y'_i - \bar{y}') \end{aligned} \quad (14)$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad \bar{x}' = \frac{1}{n} \sum_{i=1}^n x'_i \quad \bar{y}' = \frac{1}{n} \sum_{i=1}^n y'_i \quad (15)$$

この解を用いることで、入力スキャンと参照スキャンの粗な張り合わせが対応点探索によって求めた 3 組の対応点を用いて、[39, 40] に基づいて求められた位置姿勢の変化量を用いることで、入力スキャンと参照スキャンの粗な張り合わせが可能になる。

2. 詳細な張り合わせ処理:

先述した張り合わせ処理では、対応するキーポイントのデータだけを用いて位置姿勢の変化量を求めているため、近似解しか求めることができない。そこで、キーポイント周辺の複数のスキャン点も用いて、ICP アルゴリズム [38, 4] によって、位置姿勢の変化量の詳細解を求める。対応点の存在するキーポイント i^* のインデックス順に前後 O 個を ICP アルゴリズムに使用する点の数（以後、有効点数と呼ぶ）とする。この個数を大きい値にしすぎると、地図作成時とスキャン時で地形が変化していた場合うまく張り合わせをすることができなくなるので、ある程度小さい値を設定する。ICP アルゴリズムの初期値は、先述の粗な張り合わせ処理で求められた推定量を用いる。これにより、ICP アルゴリズムの初期値問題を解決できる。また、対応点の求められたキーポイント周辺のスキャン点のみを使用することにより、入力・参照スキャン中で共通して観測されるスキャン点を用いることができるため、ICP アルゴリズムが安定収束して、繰り返し計算の回数を減らす事ができる。

2.4 検証実験

SICK 社のレーザレンジファインダ (LMS100 : 視野 270[deg], 角度分解能 0.25[deg], 測定可能距離 20[m]) を用いて和歌山大学システム工学部 A 棟 1 階のフロア (65[m] × 45[m]) でスキャンを取得し, スキャン 66 個を用いて総スキャン点数 13003 個の地図を生成して実験に使用した. Fig. 11 は, 地図を構成する全てのスキャン点を可視化したものである.

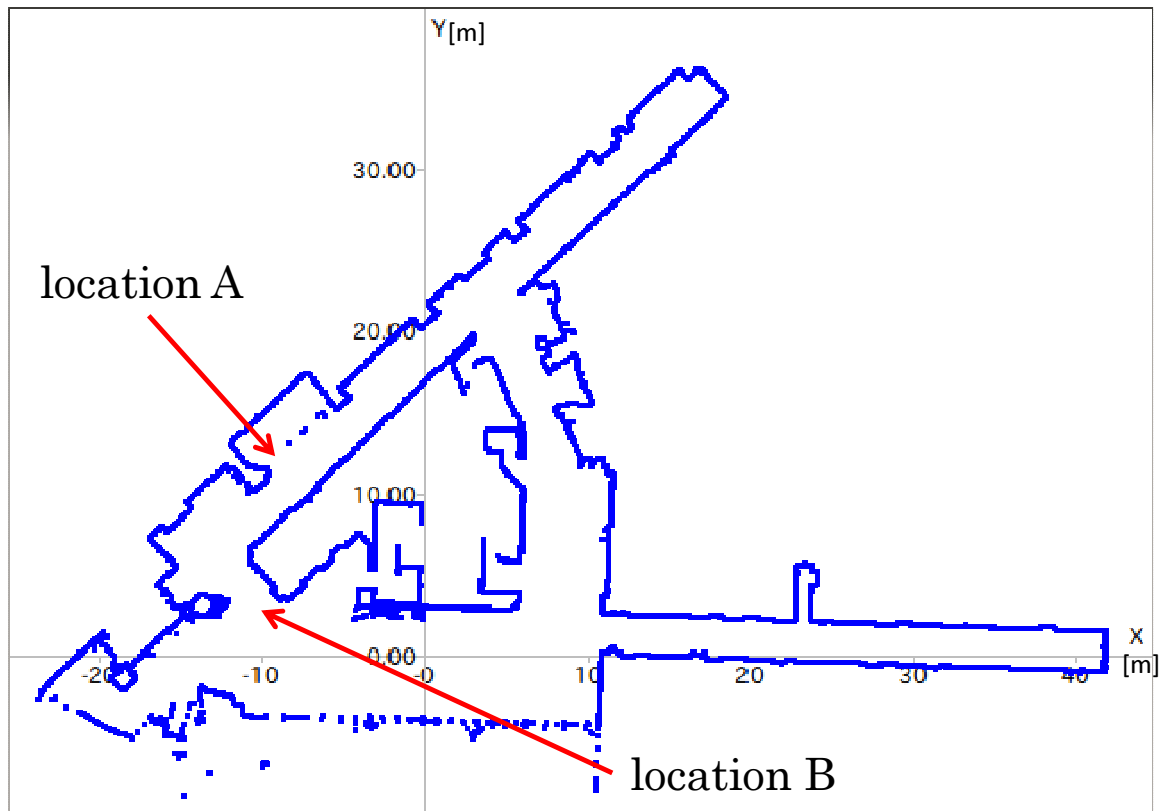


Fig. 11 The map of 1st floor in the building of Wakayama University

2.4.1 地形複雑度によるスキャンの形状特徴の判断の検証

環境地図内で, 地図作成に用いたスキャンとは別に 1482 地点のスキャンを収集し, そのスキャンと, 地形複雑度を用いて CIF-based スキャンマッチング法による自己位置推定を行った際の結果 (CIF-based スキャンマッチングに成功する・失敗するというラベル) を記録して, SVM に基づく識別器のための訓練データを作成した. 訓練データ作成時のラベルの割り振りは自己位置推定の結果を人が見て判断する.

SVM に基づく識別器を用いて, 環境の形状特徴の有無を判断する実験を行った. SVM に基づく識別器に入力する地形複雑度は, 主成分分析を用いて 5 次元に次元圧縮してから入力した. この時の累積寄

与率は 92.4% である．実験では，SVM で用いるカーネル関数として，RBF カーネルを用い，カーネルパラメータはグリッドサーチによって決定した．収集された 1482 個のデータ（地形複雑度とラベル（形状特徴の十分・不十分）の組）はラベルの数に偏りがあったため（形状特徴十分:948 個，形状特徴不十分:534 個），ランダムに形状特徴十分・不十分のデータを 500 個ずつ選び SVM に基づく識別器の訓練に使用した．10 分割交差検証によって，訓練した識別器の識別率を評価したところ，識別率は 83% であった．スキャン内の環境の形状特徴の有無の判断を誤る場合に 2 つの種類がある（(1) 特徴が十分な場合でも不十分と判断する誤り，(2) 特徴が不足するにも関わらず特徴が十分と判断する誤り）．(1) が起こる状況では，形状特徴が十分あるので自己位置推定は可能であると考えられる．(2) の場合は自己位置推定に失敗してしまうことが予想される．今回の実験では (1) が起こる確率は全体の 9% で，(2) が起こる確率は全体の 8% であった．形状特徴が十分にある場合自己位置推定に成功するとみなせる．このことから，識別の成功した 83% では自己位置推定に成功し局所地図生成に用いるスキャン数が最小となる．(1) が起こる 9% では自己位置推定に成功するが局所地図生成に用いるスキャン数が最小にならない．(2) の起こる 8% では自己位置推定が失敗する．従って，環境の形状特徴の有無を判断する識別率が 83% であっても，そのような識別器を用いて自己位置推定に成功する確率は，約 $83 + 9 = 92\%$ となる．この結果から，スキャンから地形複雑度を算出し，SVM 識別器を用いてスキャン内の環境の形状特徴の十分・不十分の判断する方法は，自己位置推定法にとって有効であると考えられる．

識別器の訓練に必要な訓練データは，本研究では，人間が判断してラベル（環境の形状特徴が十分か不十分か）の付与を行った．しかし，環境により訓練データが膨大になり，ラベル付与に対するコストが増す可能性がある．ラベルの付与を自動化させることは今後の課題であるが，次のような手順で実現できると考えられる．① 地図内でランダムに多数の場所を選び，その地点において観測されるスキャンを生成する（地図データ内から 1 時刻分のスキャンデータを選択する）．② 生成したスキャンを入力データとして CIF-based スキャンマッチング法による自己位置推定を行う．③ はじめに選択した場所（位置姿勢）と推定された位置姿勢が一致しているか否かを判定して，ラベルを付与する．

2.4.2 選択的統合処理の検証

Fig. 11 に示した環境でロボットを走行させ，その環境内の 120 地点で提案手法による大域的自己位置推定を行った．その際に，各地点で選択的統合処理によって作成された局所地図を構成するスキャンの個数を調査した．Fig. 12 は，その調査結果を，縦軸を地点数，横軸をスキャンの個数としたヒストグラムとしてまとめたものを表している．このヒストグラムに表されているように，局所地図を作成せずにその地点におけるスキャンのみで大域的自己位置推定が行えたのは 79 地点で，その他の地点では

複数のスキャンを統合して局所地図を作成していることが分かる．複数のスキャンの個数としては 2 個から 28 個まで分布しており，地点に応じて局所地図作成に用いるスキャンの個数が変化していることが分かる．

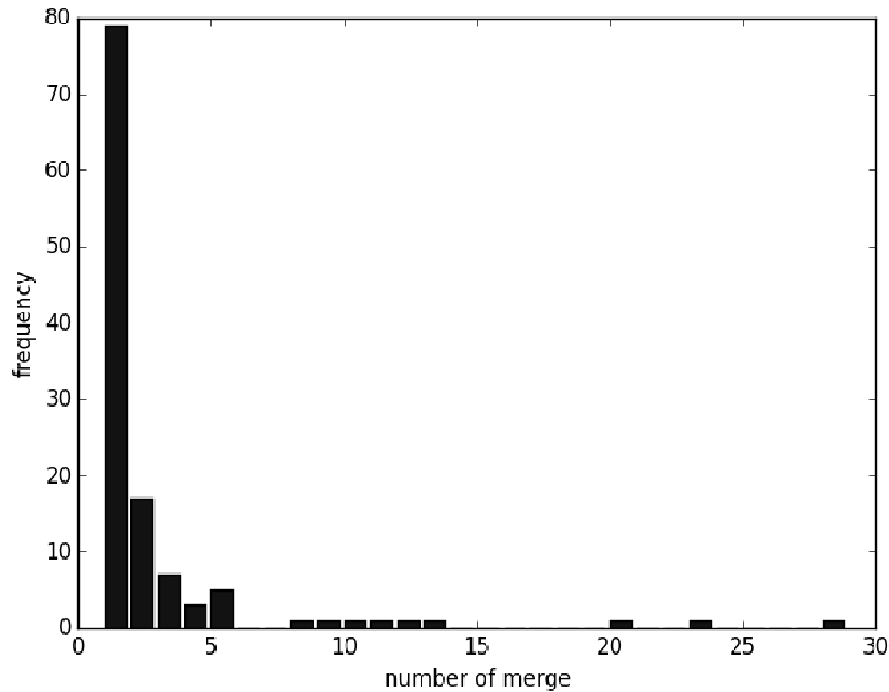


Fig. 12 Histogram of the number of scans used for generating a local map

2.4.3 提案手法による大域的自己位置推定の検証

提案手法による大域的自己位置推定の結果の例として，Fig. 11 に示した場所 A と場所 B における結果を示す．場所 A，場所 B において 1 時刻に観測されるスキャンは，それぞれ，Fig. 13(a) と Fig. 14(a) に示されている．Fig. 13(a) に示されているスキャンは，壁面に大きな凹凸のある直線的な廊下（場所 A）で取得されたもので，1 時刻のスキャンデータの中にオクルージョンがあり，形状特徴も少ない．Fig. 14(a) に示されているスキャンは，ガラス張りの大きな広間（場所 B）において取得されたもので，1 時刻のスキャンデータ中にレーザが透過する箇所が複数存在するため，不連続点を多く含み，形状特徴を得るのが難しい．これらの例で示されているような，1 時刻のスキャンのみでは大域的自己位置推定が困難な場所について，提案手法が有効であるか検証した．

場所 A において提案手法により大域的自己位置推定を行った際には，3 時刻分のスキャンを統合してできた局所地図が生成された．Fig. 13(b) は生成された局所地図を表している．また，場所 B において提案手法により大域的自己位置推定を行った際には，8 時刻分のスキャンを統合してできた局所地図が生成された．Fig. 14(b) は生成された局所地図を表している．それぞれの場所において，1 時刻のみ

のスキャン（各図 (a)）では破線で囲んだ部分にスキャン点が存在しない部分（オクルージョン）がある．一方，各図 (b) の生成された局所地図の破線で囲んだ部分には，スキャン点が存在していることが分かる．

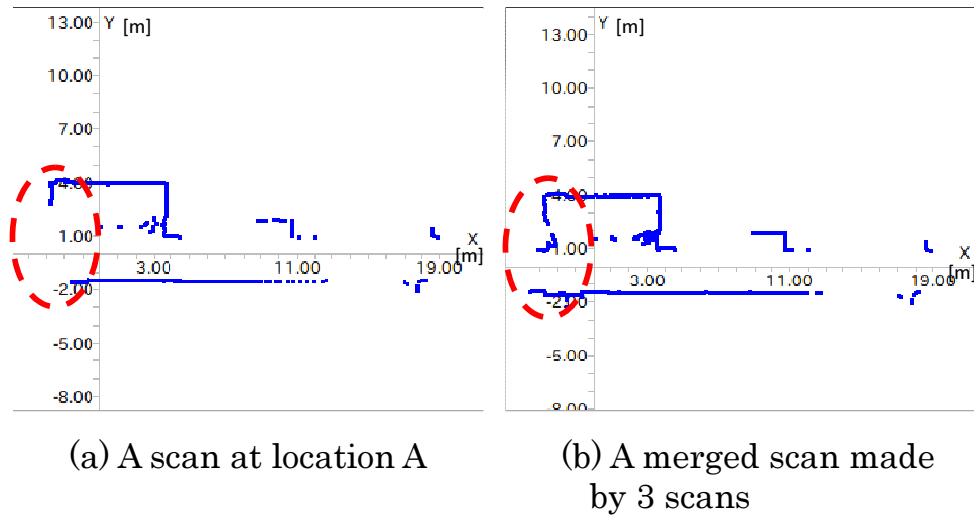


Fig. 13 The example of a raw and a merged scan at location A

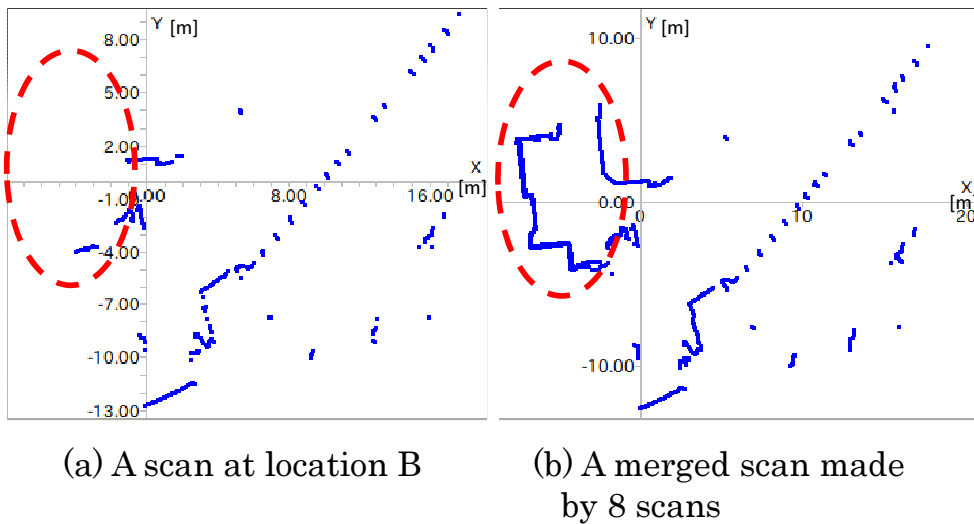


Fig. 14 The example of a raw and a merged scan at location B

Fig. 11 に示した場所 A と場所 B において，1 時刻のみのスキャンを用いて CIF-based スキャンマッチング法により大域的自己位置推定を行い，推定された自己位置に従って環境地図にスキャンを重ね合わせた結果を Fig. 15, Fig. 16 に示す．これらの図内の黒色の点は環境地図を表し，図内の破線で囲まれた部分に，1 時刻分のスキャンが赤色の点の集合として表されている．また，特徴量マッチングにおいて最も一致度の高いキーポイントの組み合わせを緑色の点で示している．

場所 A，場所 B においては，1 時刻分のスキャンだけでは大域的自己位置推定に失敗している．



Fig. 15 A result of CIF-based scan matching without merging at location A



Fig. 16 A result of CIF-based scan matching without merging at location B

Fig. 17, Fig. 18 は、それぞれ、Fig. 11 に示した場所 A と場所 B において、提案手法により Fig. 13(b), Fig. 14(b) の局所地図を生成した後、これらの局所地図と Fig.11 に示した環境地図との位置合わせを行った結果を表している。これらの図内の黒色の点は環境地図を表し、図内の破線で囲まれた部分に、生成された局所地図が赤色の点の集合として表されている。また、特徴量マッチングにおいて最も一致度の高いキーポイントの組み合わせを緑色の点で示している。

この結果を見ると、2つの場所で、それぞれ、3時刻分と8時刻分のスキャンから生成した局所地図と環境地図とで正しく位置合わせが行えており、提案手法によって正しく大域的自己位置推定が行えていることが分かる。

Fig. 19, Fig. 20 は、それぞれ、Fig. 11 内に示した場所 A, 場所 B において、ロボットが自己位



Fig. 17 A result of our method at location A



Fig. 18 A result of our method at location B

置を見失った（ロボット誘拐問題が発生した）と想定して、複数時刻のスキャンを入力して MCL 法によって大域的自己位置推定を行った結果を表している．初期状態で、環境地図内に 5000 個のパーティクルをランダムに散布し、スキャンを順次入力後、散布されたパーティクルが 1 か所に収束するまでに入力したスキャンの数を調査した．Fig. 19 の破線で囲んだ部分に示すように、場所 A では、大域的自己位置推定に失敗し、パーティクルが 1 か所に収束するまでに入力したスキャン数は 4 であった．Fig. 20 の破線で囲んだ部分に示すように、場所 B では大域的自己位置推定に成功し、パーティクルが収束

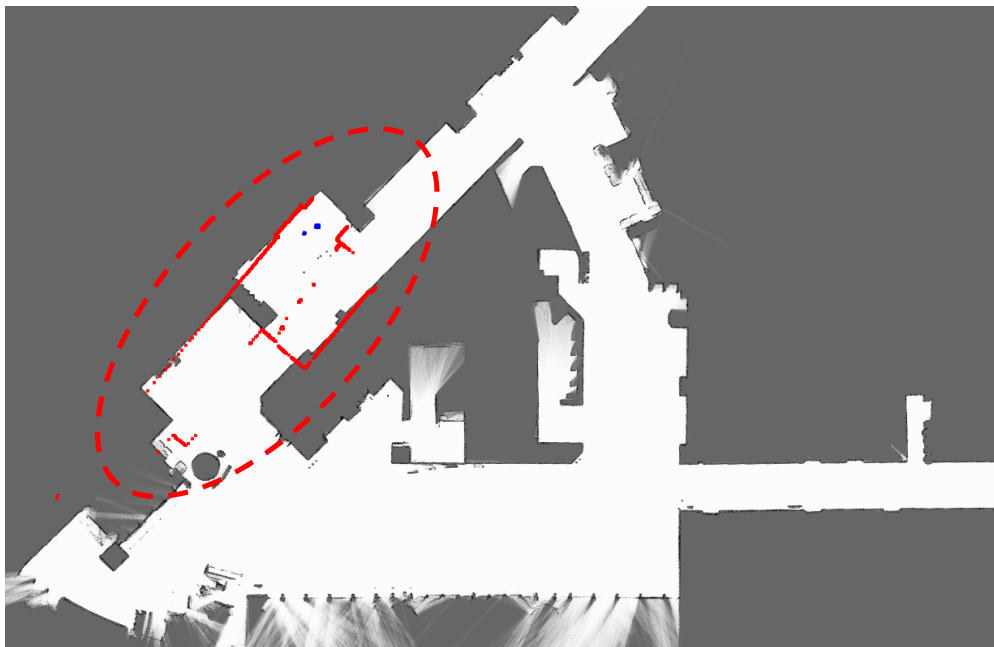


Fig. 19 A result of MCL method at location A

するのに入力したスキャン数は，14 であった．



Fig. 20 A result of MCL method at location B

これらの結果を見れば分かるように，この2つの場所では，提案手法を用いた場合の方がMCL法よりも自己位置推定に要するスキャン数が少なく済むことが分かる．また，MCL法で大域的自己位置推定に失敗するような場合でも，提案手法では自己位置推定に成功することが分かる．

提案手法によって位置合わせを行った結果得られた位置姿勢と真値^{*8}との比較を行った．Table 1 は，Fig. 17，Fig. 18 に示した位置合わせの結果から得られた位置姿勢と真値との誤差を表している．

Table. 1 Difference between actual value and estimated value

	$\Delta\tilde{x}[m]$	$\Delta\tilde{y}[m]$	$\Delta\tilde{\theta}[deg]$
location A	+0.053	+0.02	+0.5
location B	-0.08	-0.29	+0.4

この結果から分かるように，提案手法で推定した位置姿勢は誤差が小さいことが分かる．

2.4.4 未学習な環境における提案手法の有効性の検証

ある環境のデータで訓練した識別器が他の環境のデータに対して適用可能か検証するために，和歌山大学システム工学部 A 棟 7 階のフロアでスキャンを取得し，スキャン 150 個を用いて総スキャン点数 25018 個の地図を生成して実験に使用した．Fig. 21 は，地図を構成する全てのスキャン点を可視化したものである．

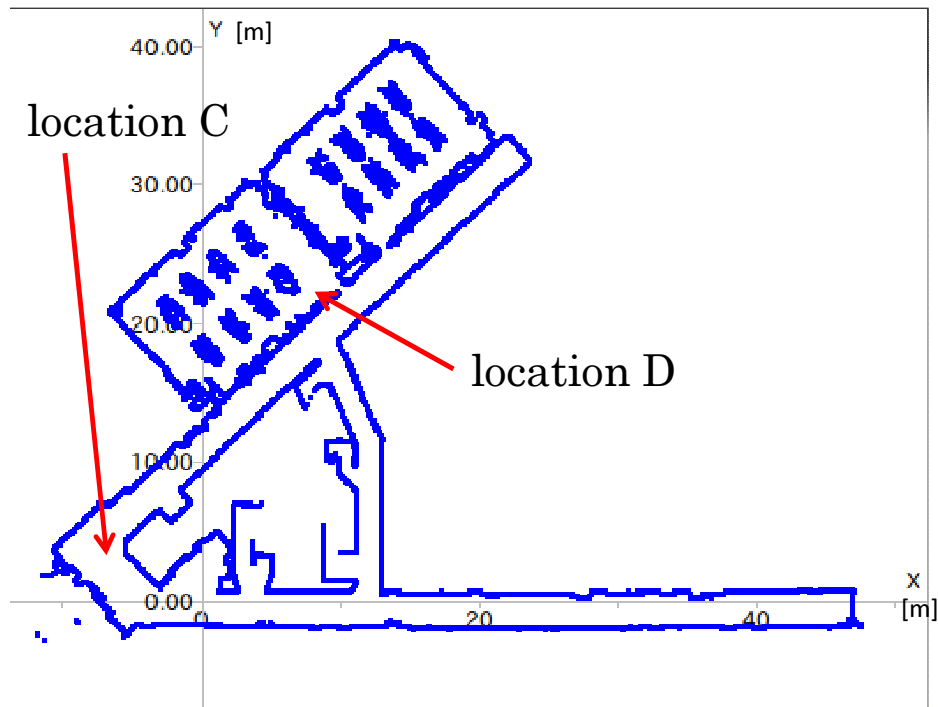


Fig. 21 The map of 7th floor in the building of Wakayama University

Fig. 21 内の場所 C と場所 D において，提案手法により大域自己位置推定を行う．Fig. 22(a)，Fig.

^{*8} 地図とスキャンの位置合わせを手動で行い，その位置姿勢を真値とした．

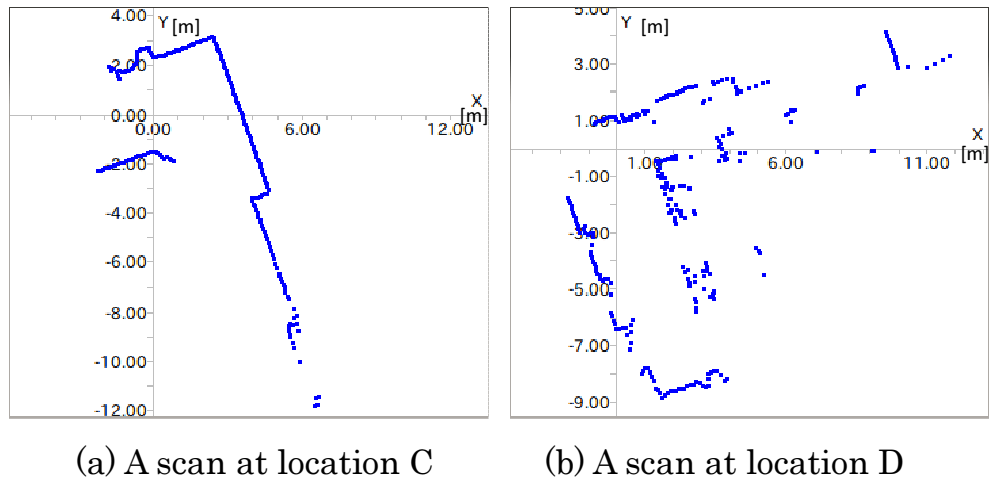


Fig. 22 Two examples of raw scans

22(b) は、それぞれ、場所 C（廊下の曲がり角）、場所 D（フロア内の一室の入り口）において取得されたスキャンを表している。いずれのスキャンも、識別器を訓練する時に用いたデータには存在しなかったスキャンである。場所 C、場所 D において、提案手法によって大域的自己位置推定した結果を Fig.23 に示す。この図内の破線で囲まれた部分に、生成された局所地図が示されている。

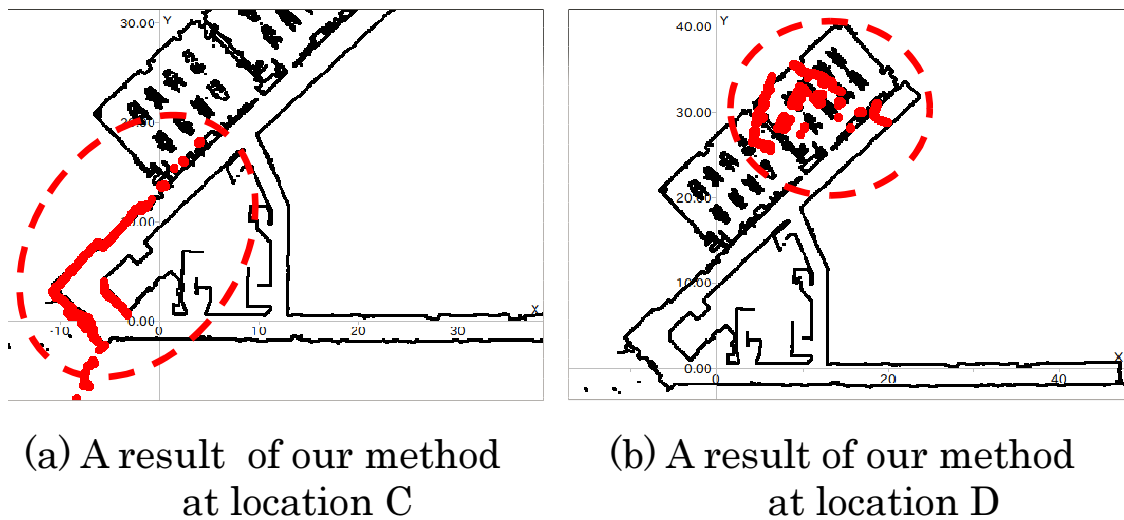


Fig. 23 Two results of our method at location C and D

Fig. 23(a) に示すように、場所 C では、4 時刻分のスキャンから局所地図を作り、自己位置推定が行われ、正しい結果が得られた。一方、Fig. 23(b) に示すように、場所 D では、自己位置推定に用いられたスキャン数は 1 つであった。このスキャンに対しては形状特徴が不十分なのにもかかわらず、識別器には形状特徴が十分と判断された。この結果から、識別器訓練時には存在しなかった形状のスキャンに対しては統合の必要・不要の判断を誤る可能性があり、最終的に大域的自己位置推定を誤ってしまうこ

とが分かる.

2.5 まとめ

本研究では、スキヤンの地形の特徴を表す“地形複雑度”を提案し、それに基づいてそのスキヤンに環境の形状特徴が十分に含まれているか否かを判断して、用いるスキヤンの数を変化させて局所地図を生成する方法“選択的統合処理”を提案した。また、生成した局所地図を使用して大域的スキヤンマッチングを行う方法を提案した。実環境の地図を用いて大域的自己位置を行う実験結果から、地形複雑度に基づいてスキヤンに含まれる形状特徴が十分か否かを RBF カーネルを用いた SVM に基づく識別器により判断できること、選択的統合処理により位置合わせに用いるスキヤンの数を限定せずに局所地図を生成できること、また、その局所地図を用いて大域的自己位置を頑健に行えることを示した。

比較実験により、提案手法を用いることで、MCL 法を用いるよりも大域的自己位置推定に用いるスキヤン数が少なくなることを示した。また、レーザが透過して特徴を得ることが難しいガラス張りの環境や、得られる特徴が少ない廊下のような環境においても、提案手法を用いることで、正しく大域的自己位置推定を行えることを確認した。提案手法は、1 時刻のスキヤン内では位置合わせに有効な特徴が少ないが、過去に遡った複数の時刻におけるスキヤン内に位置合わせに有効となる特徴が存在する環境では、有効である。しかし、過去のスキヤンをいくら遡っても特徴が存在しないような環境（例えばまっすぐな廊下が永遠と続く環境）では、提案手法による大域的自己位置推定は失敗することが予想されるが、これに対応することは今後の課題である。また、識別器を訓練する時に用いたデータ中に存在しない形状が現れた場合、識別器が判断を誤る可能性があるため、識別器は、多様な形状を含む環境で取得したデータを用いて訓練する必要がある。今後の展望として、提案手法を 2 次元レーザスキヤンを用いた大域的自己位置推定に用いるだけでなく、3 次元レーザスキヤンを用いた大域的自己位置推定の問題に適用することが考えられる。

3 LaserVAE による特徴量生成とその特徴量に基づいた大域自己位置推定

本研究では、移動ロボットの走行環境における自己位置推定に適した特徴量を自動で生成するために、variational autoencoder (VAE)[22] を独自に拡張した LaserVAE を提案する。また、LaserVAE によって生成した特徴量を利用した自己位置推定法を提案する。VAE は二つの機能 (encoder, decoder と呼ばれる) を持つニューラルネットワークである。encoder は入力データを圧縮する機能を持ち、decoder は圧縮されたデータから元の入力データを復元するという機能を持つ。ニューラルネットワークの学習後、encoder と decoder は別々に利用できる。我々の提案する LaserVAE は、通常の VAE にスキャン内のステップエッジ部分を検出する機能を有するネットワーク (step-edge classifier) を追加したもので、スキャンを圧縮して特徴量を生成する機能と、生成した特徴量からスキャンの再構成を行う機能を有するマルチタスクニューラルネットワークとなっている。LaserVAE は静的な環境だけでなく動的な環境においても使用可能な特徴量を生成することができる。また、特徴量のみを環境地図として保持することで、生のスキャンを保持する必要がなく、地図のデータ量を削減できるという特徴を持つ。本論文では、実際の屋内・屋外環境において 2 次元レーザスキャナを用いた大域的自己位置推定の実験を通して、提案する LaserVAE の有効性を示す。

3.1 LaserVAE

本研究では、ロボットの走行環境毎に、環境の形状を表す特徴量を 2 次元レーザスキャナから得られるスキャンから自動生成する手法“LaserVAE”を提案する。LaserVAE は、ロボットの走行環境において収集したスキャンを用いて、スキャンから特徴量を生成する機能 (encoder によって作られている) と、その特徴量からスキャンを再構成する機能 (decoder と step-edge classifier によって作られている) を持つ。また、encoder によって生成された特徴量を地図として利用することで、自己位置の検索を高速に行うことができる。さらに、decoder と step-edge classifier を用いて特徴量からスキャンを再構成し、スキャンマッチングを行うことで、大域的自己位置推定を実現することができる。

本研究で使用する 2 次元レーザスキャナは、周囲 270[deg] 角度分解能 0.5[deg] で測距し、その測距情報 (スキャン) を $^{raw}\mathbf{P} = (^{raw}p_1, \dots, ^{raw}p_D)$ (ここで、 D はスキャンを構成する総点数) として出力する。このベクトルの要素の順番がレーザの照射方向の順番を表し、各要素の値はその照射方向で測距された距離値を表している。また、世界座標系に対する自己位置 $^{global}\mathbf{q} = (\Delta\tilde{x}, \Delta\tilde{y}, \Delta\tilde{\theta})$ は、2 次元平面上の x 座標、 y 座標とロボットの向きによって表す。また、スキャン取得時の世界座標系に対する自己位置 $^{map}\mathbf{q}$ と encoder によって生成した特徴量 $^{map}\mathbf{z}$ の組合せをリスト構造として保存したもの $Map = \{(^{map}\mathbf{z}^1, ^{map}\mathbf{q}^1), \dots, (^{map}\mathbf{z}^M, ^{map}\mathbf{q}^M)\}$ を環境地図とし、予めこのような環境地図を作成しておくものとする (ここで、 M は地図内の特徴量と自己位置の組の総数である)。

ある時刻に取得したスキャンの計測値の中には、移動物体やオクルージョンの影響により、計測した距離の値が急激に変化するステップエッジが含まれている場合がある。このようなステップエッジがあるスキャンに対して、値の連続性を仮定する通常の VAE の decoder により再構成すると、本来ステップエッジになる部分 (Fig.24(b) 丸印部分) が平滑化されて、再構成時にステップエッジ部分を再現できない。LaserVAE は、通常の VAE にステップエッジを検出する機能を追加し、ステップエッジを含むスキャンを再構成できるように改良した。

3.1.1 LaserVAE のネットワーク構造

LaserVAE のネットワーク構造の説明の前に通常の VAE について説明する。VAE は、AutoEncoder(AE) の発展形として登場した。AE は教師なし学習手法の一つで、入力データ \mathbf{x} に対して次元削減を行い潜在変数 \mathbf{z} に変換するニューラルネットワーク encoder と、潜在変数 \mathbf{z} から入力データを復元するニューラルネットワーク decoder を学習する。ニューラルネットワークが 1 層の場合を式で表すと以下のようなになる。

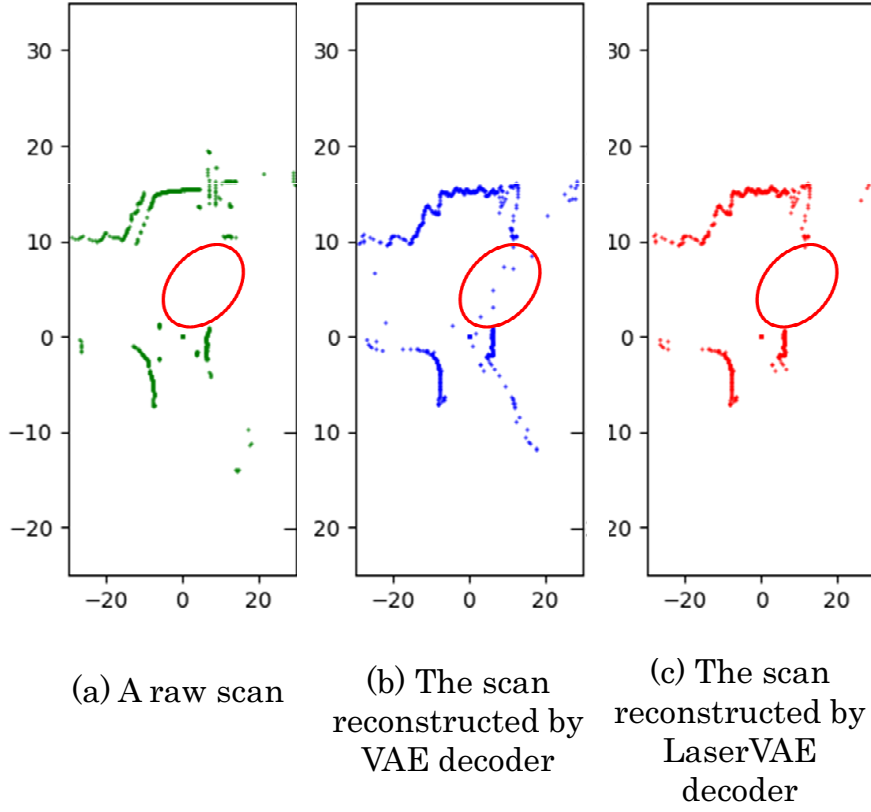


Fig. 24 A raw scan and the reconstructed scans

$$\hat{\mathbf{x}}(\mathbf{x}) = \widehat{\mathbf{W}}(\mathbf{W}\mathbf{x} + \mathbf{b}) + \hat{\mathbf{b}} \quad (16)$$

$$\mathcal{L}_{\text{AE}} = \sum_{i=1}^N (\mathbf{x}_i - \hat{\mathbf{x}}_i(\mathbf{x}_i))^2 \quad (17)$$

ここで、 \mathbf{W} は重みと呼ばれる行列で、 \mathbf{b} はバイアスと呼ばれるベクトルで、重みとバイアスを用いて入力データを別の次元へ変換している。 $\mathbf{W}\mathbf{x} + \mathbf{b}$ が encoder の出力潜在変数 \mathbf{z} に相当する。decoder の出力 $\hat{\mathbf{x}}$ が入力データ \mathbf{x} に近づくように \mathcal{L}_{AE} を最小化するように学習を行う。

VAE は、AE の潜在変数 \mathbf{z} に確率分布を仮定した点異なる。 \mathbf{z} に確率分布を仮定することによって学習済みのニューラルネットワークは生成モデルとして活用可能となる。

Fig.25 に示すように LaserVAE は、従来の VAE の encoder と decoder に加え、再構成したスキャンデータ上のステップエッジを検出する step-edge classifier から構成されるマルチタスクニューラルネットワークである。

各ネットワークの構造を Fig.26 に示す。encoder は畳み込みを行う convolution 層が 3 層と全結合が 1 層の構造になっている。decoder は全結合層が 1 層と畳み込みの逆の処理に相当する deconvolution 層が 3 層という構造になっている。step-edge classifier は全結合のニューラルネットワークとなってい

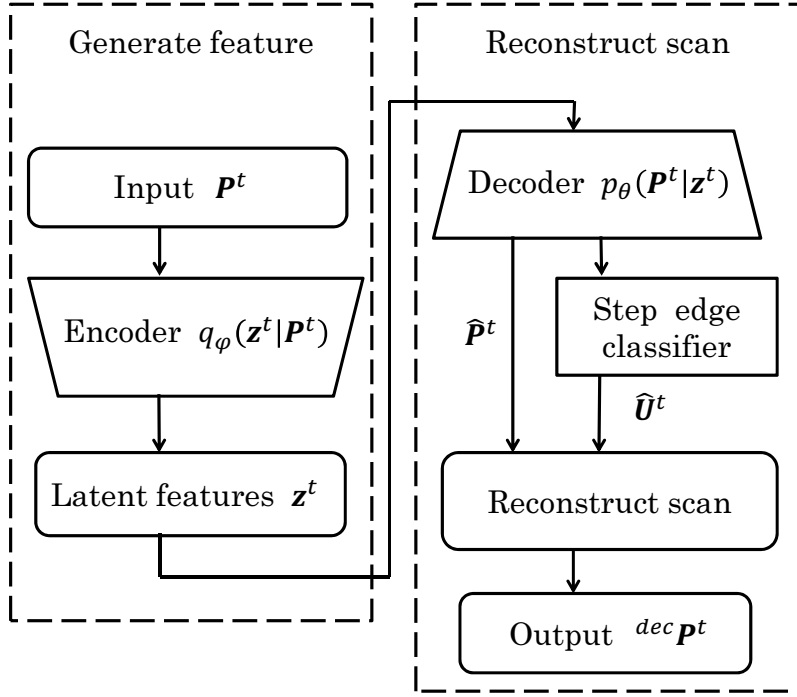


Fig. 25 Overview of LaserVAE

る．ここで、 K は特徴量の次元数で、本論文では $K=20,50,120$ の 3 種類で学習を行った．

encoder への入力ベクトル \mathbf{P}^t は 2 次元レーザスキャナで取得したスキャン $^{raw} \mathbf{P}^t$ を最大測定範囲 r_{max} で割った値 ($\mathbf{P}^t = \frac{^{raw} \mathbf{P}^t}{r_{max}}$) を用いる (今回は $r_{max} = 50$)．

3.1.2 LaserVAE の学習方法

LaserVAE では、encoder と decoder の学習後、step-edge classifier の学習を行う．まず、訓練データとして用意したスキャン $\mathbf{P}^t (t = 1, \dots, N)$ を用いて、通常の VAE の学習を行い encoder および decoder のパラメータ（各層間の重み）を求める．ここで、 N は学習に用いるスキャンの総数である．

通常の VAE は、入力ベクトル \mathbf{P}^t を低次元の特徴ベクトル \mathbf{z}^t に圧縮する encoder $q_\phi(\mathbf{z}^t | \mathbf{P}^t)$ および、 \mathbf{z}^t を元の入力ベクトル $\hat{\mathbf{P}}^t$ に再構成する decoder $p_\theta(\hat{\mathbf{P}}^t | \mathbf{z}^t)$ から構成されている．encoder $q_\phi(\mathbf{z}^t | \mathbf{P}^t)$ は \mathbf{P}^t が得られた時にその \mathbf{P}^t を生成した \mathbf{z}^t を推論する確率分布を表し、decoder $p_\theta(\hat{\mathbf{P}}^t | \mathbf{z}^t)$ は \mathbf{z}^t から高次元ベクトル $\hat{\mathbf{P}}^t$ を生成する確率分布を表している．そして、ネットワークのパラメータ θ および ϕ (各層における重みとバイアスの組) を、汎化性を向上するための罰則付きの二乗損失を最小化するように学習する．

$$\mathcal{L}_{VAE} = \sum_{t=1}^N (\mathbf{P}^t - \hat{\mathbf{P}}^t)^2 - \lambda \text{MMD}(q_\phi(\mathbf{z}^t | \mathbf{P}^t) || p(\mathbf{z}^t)) \quad (18)$$

ここで、 λ は、罰則項の重要度を表す係数で、 $\text{MMD}()$ は、encoder $q_\phi(\mathbf{z}^t | \mathbf{P}^t)$ と一様分布 $p(\mathbf{z}^t)$ との

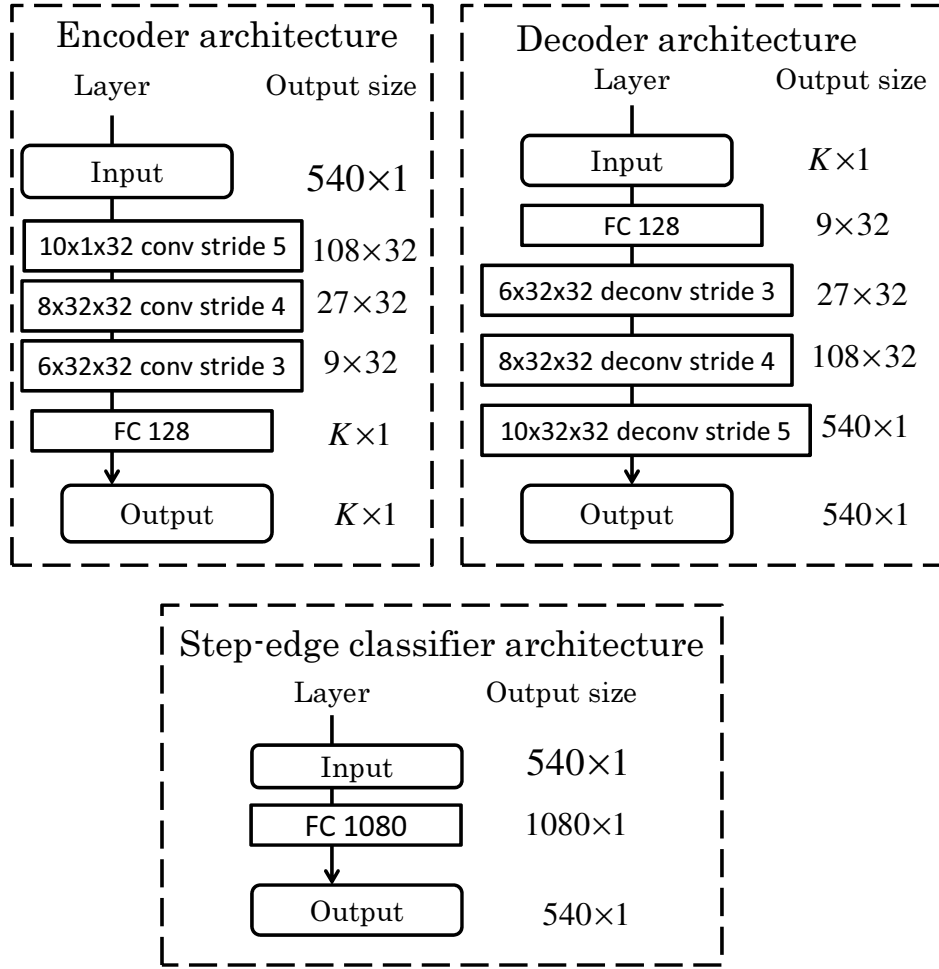


Fig. 26 Network architecture of LaserVAE. The upper left is the encoder network, the upper right is the decoder network and the under is the step-edge classifier network.

分布間距離 maximum mean discrepancy (MMD) に対応する罰則項である。MMD はカーネル関数を用いて変換した特徴空間内の二つの分布の平均値を比較することで二つの分布の類似度を求める方法である。

このように VAE では、教師データなしで入力ベクトル \mathbf{P}^t から低次元の特徴ベクトル \mathbf{z}^t を生成する encoder および \mathbf{z}^t を入力ベクトル $\hat{\mathbf{P}}^t$ に再構成する decoder を獲得することができる。

次に、求められた encoder および decoder のパラメータを用いて、step-edge classifier を学習する。スキャン \mathbf{P}^t 上のステップエッジの箇所に 0, それ以外を 1 とするラベル付けした学習データ $\mathbf{U}^t = (U_1^t, \dots, U_D^t)$ を用意し、decoder により再構成した $\hat{\mathbf{P}}^t$ 上の各点においてステップエッジか否かの識別結果 $\hat{\mathbf{U}}^t$ を出力する識別器を以下の損失関数を最小化することにより学習する。

$$\mathcal{L}_{\text{SEC}} = \sum_{t=1}^N |\mathbf{U}^t - \hat{\mathbf{U}}^t| \quad (19)$$

ここで、ステップエッジのラベル U^t は式 (20),(21) に基づいて、自動的にラベル付けしている。

$$m^t = \frac{1}{D-1} \sum_{j=2}^D |p_j^t - p_{j-1}^t| \quad (20)$$

$$U_j^t = \begin{cases} 1 & (|p_j^t - p_{j-1}^t| < m^t) \\ 0 & (|p_j^t - p_{j-1}^t| > m^t \text{ or } p_j^t > r_{max}) \end{cases} \quad (21)$$

スキャン内の連続する 2 点間の距離の平均値 m^t を算出し、スキャン内の要素 p_j^t に対して、ひとつ前の要素 p_{j-1}^t との差 $|p_j^t - p_{j-1}^t|$ が m^t よりも大きいもしくは 2 次元レーザースキャナの最大測定範囲 r_{max} より大きい場合はステップエッジ部分とみなしている。

3.1.3 LaserVAE を用いた特徴量生成とスキャンの再構成

スキャンから特徴量を生成する際には、学習後の encoder ヘスキャン \mathbf{P}^t を入力すると、そのスキャンの特徴量 \mathbf{z}^t を生成できる。特徴量からスキャンを再構成する際には、学習後の decoder と step-edge classifier を用いる。学習後の decoder に特徴量 \mathbf{z}^t を入力すると、再構成前のスキャン $\hat{\mathbf{P}}^t$ が生成される。その再構成前のスキャンを学習後の step-edge classifier に入力すると、ステップエッジ部分 \hat{U}^t を検出して、 $\hat{\mathbf{P}}^t$ 内の各点 \hat{p}_j^t に対して式 (22) に従ってステップエッジ部分を復元したスキャンを再構成できる。この時、decoder から出力された $\hat{\mathbf{P}}^t$ の値の値域は 0 ~ 1 となっているので、 r_{max} を乗算することでスキャンを再構成している。

$${}^{dec}\hat{p}_j^t = \begin{cases} r_{max}\hat{p}_j^t & (\hat{U}_j^t \geq 0.5) \\ 0 & (\hat{U}_j^t < 0.5) \end{cases} \quad (22)$$

さらに、式 (23)~(25) により孤立点 (スキャン内の各点について、その前後の点から離れた位置にある点) を除去し、再構成されたスキャン ${}^{dec}\mathbf{P}^t$ を得ることができる。ここで ξ は 2 次元レーザースキャナの角度分解能で、 λ は孤立点を検出するためのしきい値である。今回、 ξ は 0.5[deg] で、 λ は 0.87[m] とした。また、 d_j^t は再構成されたスキャン内の連続する 2 点 (${}^{dec}\hat{p}_j^t, {}^{dec}\hat{p}_{j+1}^t$) 間の距離を表し、 ${}^{dec}\hat{p}_j^t$ がその前後の点 (${}^{dec}\hat{p}_{j-1}^t, {}^{dec}\hat{p}_{j+1}^t$) と eth_j^t よりも離れている場合孤立点とみなし、除去する。

$$d_j^t = \sqrt{{}^{dec}\hat{p}_j^t{}^2 + {}^{dec}\hat{p}_{j+1}^t{}^2 - 2{}^{dec}\hat{p}_j^t {}^{dec}\hat{p}_{j+1}^t \cos \xi} \quad (23)$$

$$eth_j^t = \frac{\lambda}{r_{max}} {}^{dec}\hat{p}_j^t \quad (24)$$

$${}^{dec}p_j^t = \begin{cases} {}^{dec}\hat{p}_j^t & (d_{j-1}^t < eth_j^t \text{ and } d_j^t < eth_j^t) \\ 0 & (else) \end{cases} \quad (25)$$

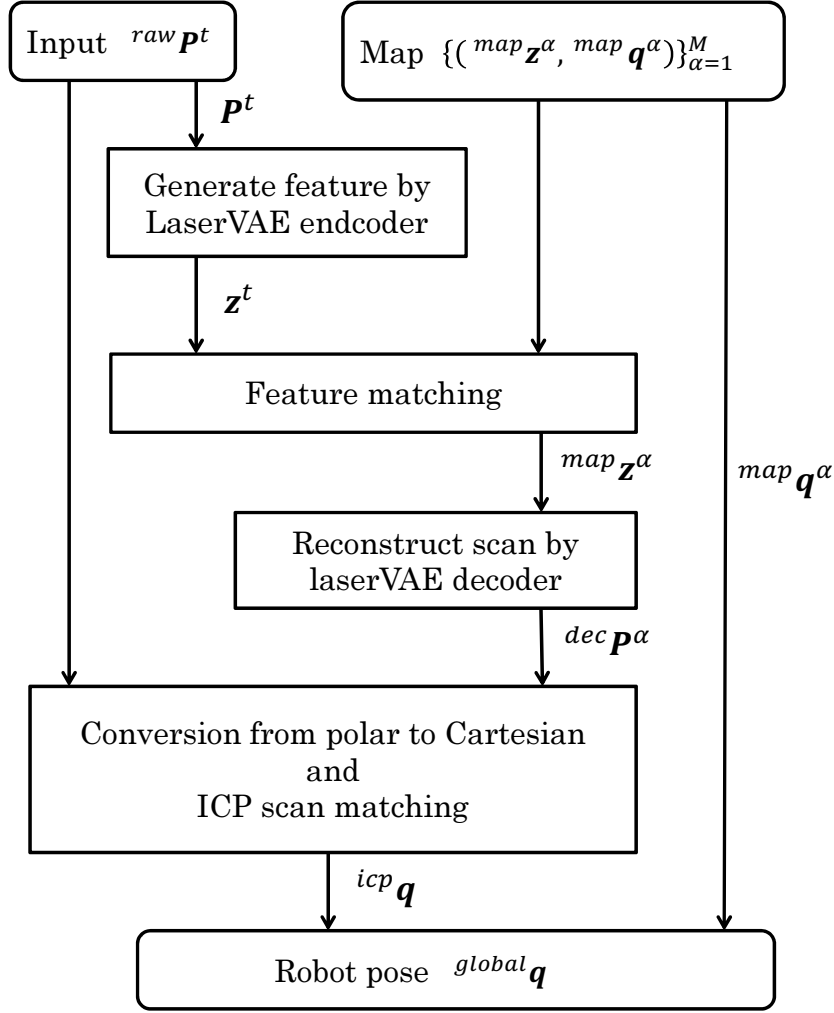


Fig. 27 Overview of our global self-localization method

3.2 大域的自己位置推定

本研究で行う自己位置推定の流れを Fig.27 に示す．時刻 t において取得したスキャン $raw \mathbf{P}^t$ に対して，encoder による特徴量生成を行い，得られた特徴量 \mathbf{z}^t と地図内の特徴量 $\{\mathit{map} \mathbf{z}^1, \dots, \mathit{map} \mathbf{z}^M\}$ の照合を行う．照合には特徴量間のユークリッド距離を用いて，最も一致する (距離の短い) 特徴量 $\mathit{map} \mathbf{z}^\alpha$ を発見する．

$$\alpha = \arg \min_{\alpha} \sqrt{\sum_{k=1}^K (z_k^t - \mathit{map} z_k^\alpha)^2} \quad (26)$$

ここで， K は encoder によって得られる特徴量の次元数である．このようにして生成した特徴量 $\mathit{map} \mathbf{z}^\alpha$ に対して decoder によりスキャンの再構成を行い，再構成されたスキャン $\mathit{dec} \mathbf{P}^\alpha$ と現在のスキャン $raw \mathbf{P}^t$ で位置合わせを行う．位置合わせを行うために， $raw \mathbf{P}^t$ を式 (27),(28) に従って 2 次元座

標上の点群 ${}^{raw}_{pc}\mathbf{P}^t$ に変換する.

$$\theta_j = \frac{j - 90}{360} \pi \quad (27)$$

$${}^{raw}_{pc}\mathbf{P}_j^t = \begin{pmatrix} {}^{raw}p_j^t \cos \theta_j \\ {}^{raw}p_j^t \sin \theta_j \end{pmatrix} \quad (28)$$

同様に, ${}^{dec}\mathbf{P}^\alpha$ も点群 ${}^{dec}_{pc}\mathbf{P}^\alpha$ へ変換し, 得られた二つの点群 (${}^{dec}_{pc}\mathbf{P}^\alpha, {}^{raw}_{pc}\mathbf{P}^t$) を入力として Iterative Closest Point(ICP)[3, 4] を適用する. 2 つの点群間で仮の対応点 (最近傍点) を設定し, 片方の点群に対して回転・並進を繰り返し, 対応点間の距離の総和を最小化する最適化計算を行うことで 2 つの点群 (今回はスキャン ${}^{dec}_{pc}\mathbf{P}^\alpha, {}^{raw}_{pc}\mathbf{P}^t$) 間の相対的な位置姿勢 ($\Delta q_x, \Delta q_y, \Delta q_\theta$) を推定する.

式 (29) を用いて, 求められた相対的な位置姿勢と特徴量に紐づいている位置姿勢 ${}^{map}\mathbf{q}^\alpha = ({}^{map}q_x^\alpha, {}^{map}q_y^\alpha, {}^{map}q_\theta^\alpha)$ から, 大域的な自己位置 ${}^{global}\mathbf{q}$ を求める.

$${}^{global}\mathbf{q} = \begin{pmatrix} {}^{map}q_x^\alpha + \Delta q_x \cos \Delta q_\theta - \Delta q_y \sin \Delta q_\theta \\ {}^{map}q_y^\alpha + \Delta q_x \sin \Delta q_\theta + \Delta q_y \cos \Delta q_\theta \\ {}^{map}q_\theta^\alpha + \Delta q_\theta \end{pmatrix} \quad (29)$$

3.3 検証実験

今回の実験でロボットが走行する環境を Fig.28 に示す．屋内，屋外 (短距離)，屋外 (長距離) 環境の合計 3 種類の環境でスキャンを取得して，LaserVAE の学習を行い，LaserVAE により作成した特徴量を用いた地図データを作成後，同じコースを走行させて自己位置推定を行った．なお，LaserVAE の学習に用いたスキャンデータの総数は，屋内，屋外 (短距離)，屋外 (長距離) 環境において，それぞれ 6018, 14660, 59861 個である．ロボットが走行中に並進移動量 0.4[m] 以上，もしくは回転変位量 10[deg] 以上移動する毎に，取得したスキャンから特徴量を生成し，その時の自己位置とのペアを環境地図情報として逐次記録する．地図作成時の自己位置は MCL 法 [12] を用いて求めた*9．また，3.3.3 での大域的自己位置推定の真値も同じく MCL 法によって求めた．

屋内環境の地図は，和歌山大学北 1 号館 1 階のフロア内の移動距離約 80[m] のコースで，特徴量と自己位置のペア 150 個で表した．屋外 (短距離) 環境の地図は，和歌山大学北 1 号館から図書館前まで急カーブを含む移動距離約 200[m] のコースで，特徴量と自己位置のペア 376 個で表した．屋外 (長距離) 環境の地図は，和歌山大学北 1 号館からキャンパスを往復する移動距離約 800[m] のコースで，特徴量と自己位置のペア 1541 個で表した．

3.3.1 特徴量の回転不変性に関する検証

LaserVAE の encoder では，スキャンを入力として特徴量生成を行うが，入力されるスキャンはその性質上，ロボットの斜め後方から順に周囲の距離データをベクトル化されている．このようにして得られたスキャンでは，ロボットの向きが変化した場合，距離データを格納する順序が異なり，同じ位置であっても生成される特徴量が一致しないことがありうる．回転不変性がない場合，特徴地図内にロボットの回転を含んだ特徴量も追加する必要があると考えられる．実験により，LaserVAE の encoder によって生成される特徴量に回転不変性があるか検証する．

Fig.29 に示した地図内の屋外 (短距離) 環境内の二つのコーナにおいて取得されたスキャンから特徴量を生成し，それを可視化した．

コーナ 1 においてロボットの回転を行い，取得されたスキャンから生成した特徴量 ($K=20$) を可視化した結果を Fig.30(a) (c) に示す．横軸が特徴量の次元数で，縦軸が各次元の値である．(a) はロボットが回転する前における特徴量で，(b) が 45[deg] の回転を行った後の特徴量そして (c) が 90[deg] の回転を行った後の特徴量である．(a) と (c) の特徴量は似たような場所に特徴が出ているが，(a) の方が全体

*9 これは一例であり，GPS が利用可能であればそちらを用いて自己位置を求めてもよい．

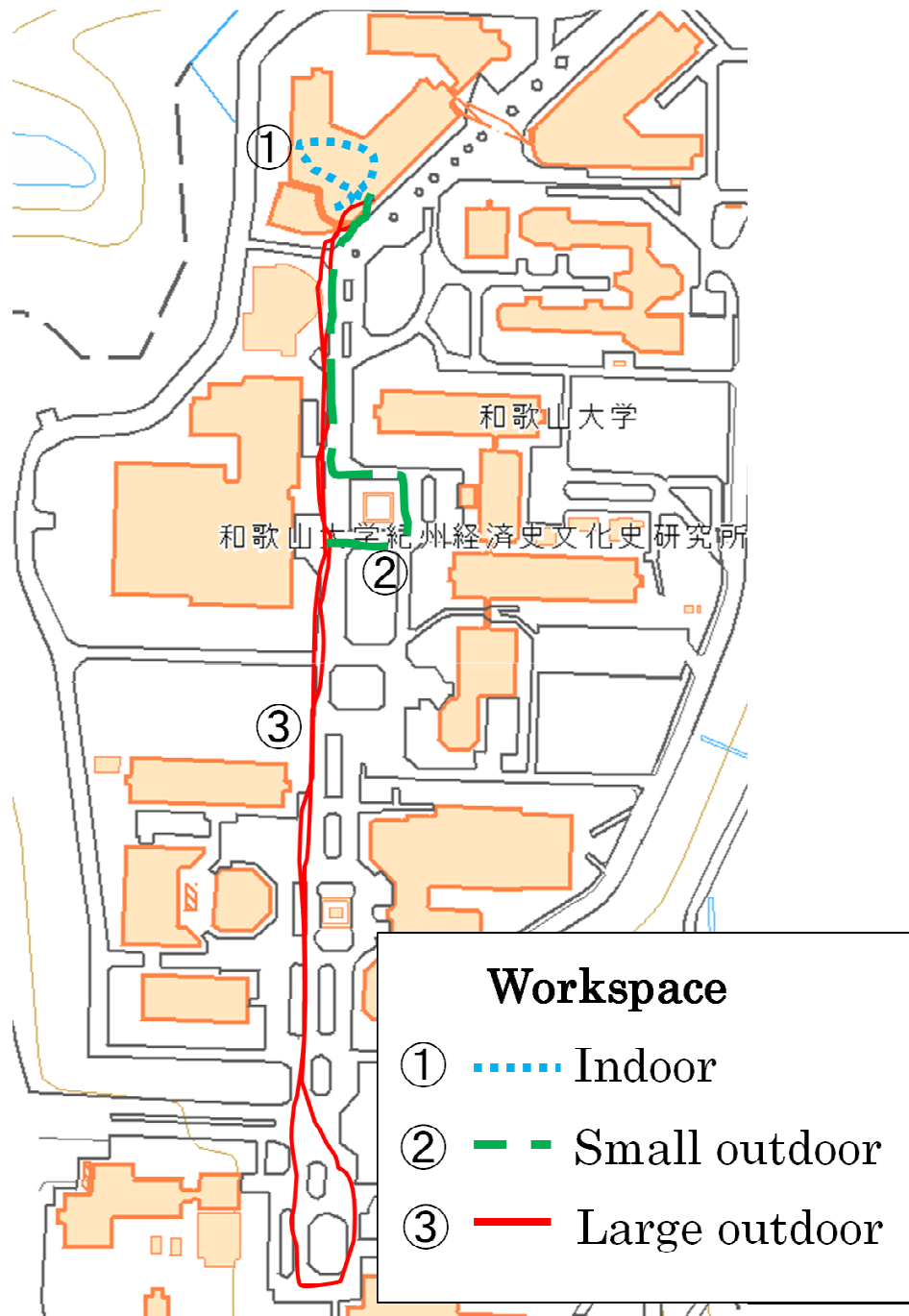


Fig. 28 Three different workspaces of our mobile robot in Wakayama University (Map: Geospatial Information Authority of Japan)

的に値が小さいため、この二つの特徴量は一致しているとはいえない。また、(b) の特徴量は他二つとは全く異なる特徴が出ており、一致しているとはいえない。

また、コーナ 2 においてロボットの回転を行い、取得されたスキャンから生成した特徴量 ($K=20$) を可視化した結果を Fig.31(a) (c) に示す。コーナ 1 と同様に左上が回転する前における特徴量で、右上が $45[\text{deg}]$ の回転を行った後の特徴量そして左下が $90[\text{deg}]$ の回転を行った後の特徴量をそれぞれ示している。これらの三つの特徴量を比較しても、コーナ 1 と同様にいずれの特徴量も一致しているとはい

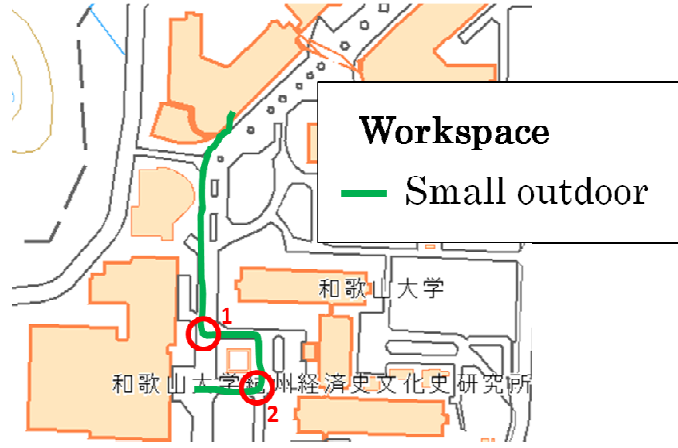


Fig. 29 Two corners in the small outdoor workspace

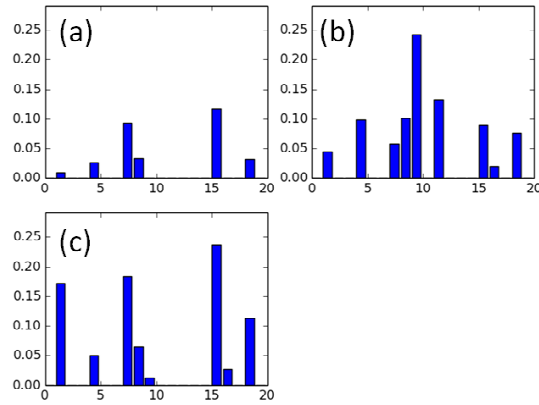


Fig. 30 The latent features at the corner1. (a) 0 [deg] rotation. (b) 45 [deg] rotation. (c) 90 [deg] rotation.

えない。

これらの結果から、同じ場所において取得されたスキャンであっても、センサの角度が異なる場合は生成される特徴量が必ずしも一致しないことが分かる。つまり、提案する LaserVAE によって生成される特徴量には回転不変性がないことが分かる。

3.3.2 特徴量からのスキャンの再構成の性能

LaserVAE の encoder を用いて作成した特徴地図 (屋外短距離) 内の各特徴量から decoder と step-edge classifier を用いて、ステップエッジを含むスキャンを再構成し、そのスキャンを特徴量に紐づいている自己位置に配置して点群地図を作成した結果を Fig.32(a) (c) に示す。 (a) は生のスキャンによって構成される点群地図で、 (b) は通常の VAE を用いてスキャンを再構成して得られた点群地図、 (c) は提案手法 LaserVAE を用いてスキャンを再構成して得られた点群地図となっている。これを見ると、提案手法 LaserVAE は、通常の VAE に比べて生のスキャンを用いて作成した点群地図に近い地図を再現

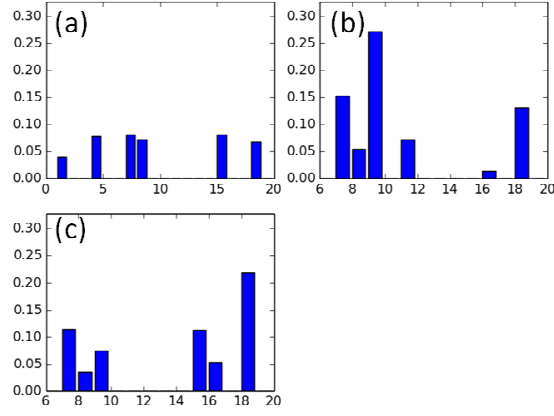


Fig. 31 The latent features at the corner2. (a) 0 [deg] rotation. (b) 45 [deg] rotation. (c) 90 [deg] rotation.

できていることが分かる．このことから，提案手法 LaserVAE の encoder によってスキャンから走行環境の地形を表す特徴量が取得できていることが分かる．

3.3.3 大域的自己位置推定の性能

提案手法による大域的自己位置推定を行い，本手法の有効性について検討する．LaserVAE で生成した特徴量と通常の VAE を用いて生成した特徴量さらに阪東らの特徴量 [18] を用いて，異なる 3 種類の環境で自己位置推定を行い，性能を比較する．LaserVAE と通常の VAE の特徴量の次元数は $K=20$, 50, 120 の 3 種類，阪東らの特徴量 [18] の量子化間隔はセンサの周囲 50[m] を 128 個の格子で区切ったものと 256 個の格子で区切った 2 種類で実験を行う．

屋内環境で自己位置推定した結果を Table.2 に示す．屋内環境で地図作成時と同じ経路を走行させ，308 か所で自己位置推定を行い，推定された自己位置と真値との誤差を計算し，その平均値を算出した．

この表を見ればわかるように，提案手法 1 の CIF 特徴量を用いた自己位置推定法では，他の手法に比べて誤差が大きくなっている．また，阪東らの特徴量 [18] を用いた場合は画像の量子化間隔に性能が左右されてはいるものの，真値との誤差は少なくなっている．これに対して，通常の VAE や提案手法 LaserVAE を用いて生成した特徴量いずれの特徴量を用いても自己位置推定の真値との誤差は同じ程度であり，特徴量の次元数によらず性能が安定していることが分かる．

屋外環境で自己位置推定した結果を Table.3, 4 に示す．屋外環境で地図作成時と同じ経路を走行させ，短距離の環境では 771 か所，長距離の環境では 3127 か所で自己位置推定を行い，推定された自己位置と真値との比較を行い，その平均値を算出した．阪東らの特徴量 [18] を用いた自己位置推定では量子化間隔にかかわらず，どちらの場合も誤差が大きくなっている．一方，通常の VAE や提案手法 LaserVAE では特徴量の次元数によらず自己位置推定の真値との誤差が小さくなっている．

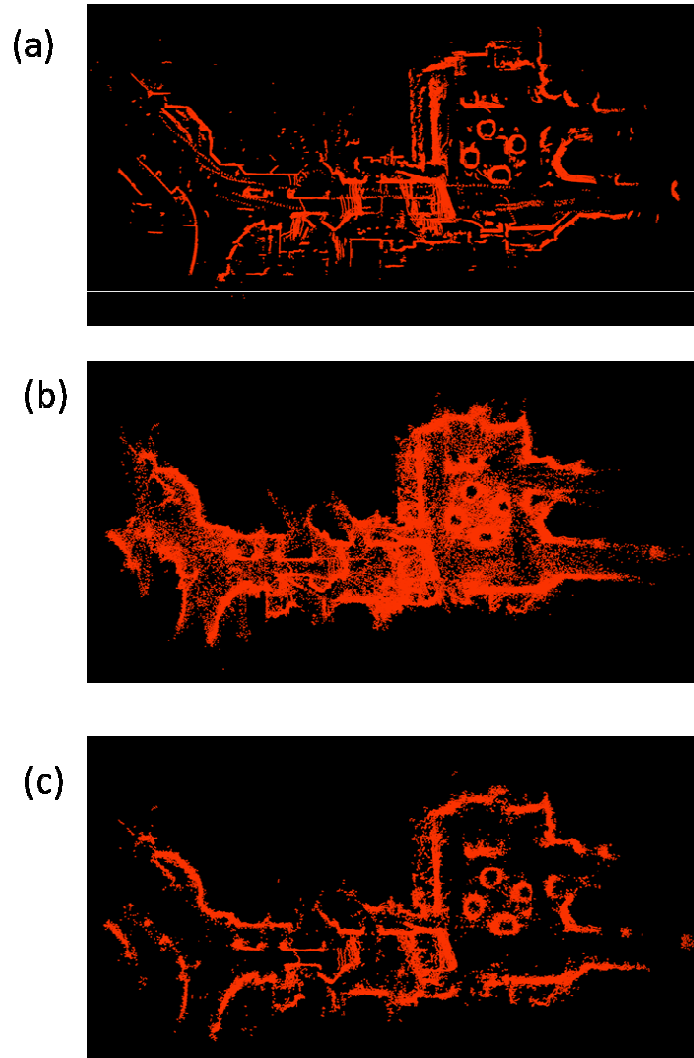


Fig. 32 (a) A map consisted of raw scans, (b) a map consisted of scans reconstructed by VAE decoder and (c) a map consisted of scans reconstructed by LaserVAE decoder

真値との半径誤差 1.58[m], 角度誤差 8.5[deg] 以内を自己位置推定成功と判断する場合の自己位置推定の成功率を Table.5 に示す.

提案手法 1 の CIF 特徴量を用いた自己位置推定法では, 成功率は 6 割程度になっている. 実験に使用した屋内環境は, 壁の素材としてガラスを使用している部分が多く, CIF 特徴量が時間的に安定して検出されなかった. それが原因で, 局所地図の作成自体に失敗することが頻繁に起こったため, 成功率が低くなったと考えられる. なお, この表で, 提案手法 1 の CIF 特徴量を用いた自己位置推定法の結果では, 屋内環境での評価しか行っていない.

また, 阪東らの特徴量 [18] を用いた場合は環境の量子化間隔に性能が左右されており, 屋内では高い成功率を得ることができていても, 人の往来や, オクルージョンが発生する屋外の環境においては成功率が低く, 環境が変化すると安定した性能を発揮することは難しいと考えられる. 阪東らの手法で用い

Table. 2 Average error of estimated robot pose in the indoor workspace

	$\Delta x[\text{m}]$	$\Delta y[\text{m}]$	$\Delta \theta[\text{deg}]$
CIF method	3.67	4.24	13.16
Bando's method[18](128)	0.80	0.91	2.8
Bando's method[18](256)	0.56	0.66	1.3
VAE($K=20$)	0.77	1.01	1.3
VAE($K=50$)	0.79	0.79	3.0
VAE($K=120$)	0.57	0.86	2.1
LaserVAE($K=20$)	0.76	1.01	1.4
LaserVAE($K=50$)	0.78	0.79	3.1
LaserVAE($K=120$)	0.57	0.87	2.2

Table. 3 Average error of estimated robot pose in the small outdoor workspace

	$\Delta x[\text{m}]$	$\Delta y[\text{m}]$	$\Delta \theta[\text{deg}]$
Bando's method[18](128)	15.19	3.72	17.0
Bando's method[18](256)	12.87	3.95	14.6
VAE($K=20$)	0.89	0.32	2.0
VAE($K=50$)	0.47	0.31	1.1
VAE($K=120$)	0.33	0.25	1.2
LaserVAE($K=20$)	0.87	0.31	2.0
LaserVAE($K=50$)	0.47	0.30	1.1
LaserVAE($K=120$)	0.32	0.26	1.2

Table. 4 Average error of estimated robot pose in the large outdoor workspace

	$\Delta x[\text{m}]$	$\Delta y[\text{m}]$	$\Delta \theta[\text{deg}]$
Bando's method[18](128)	57.96	2.75	15.6
Bando's method[18](256)	48.30	2.29	11.6
VAE($K=20$)	4.84	0.48	3.5
VAE($K=50$)	4.28	0.38	2.9
VAE($K=120$)	2.63	0.32	2.4
LaserVAE($K=20$)	4.83	0.45	3.5
LaserVAE($K=50$)	4.26	0.36	3.0
LaserVAE($K=120$)	2.62	0.31	2.2

られている特徴量は、比較的静的な環境で用いられることを想定しているため、今回の実験環境 (特に屋外の環境) のように、人の往来がある場合に自己位置推定が不安定になったと考えられる。

それに対して通常の VAE や提案手法 LaserVAE は、人の往来やオクルージョンによる影響で、時間的に不安定なスキャンデータが取得される状況であっても、安定して自己位置推定を行えることが分かる。提案手法は、学習によって環境の不変的な構造を反映した特徴量を生成できるため、人の足等のノイズの含むスキャンから生成した特徴量であっても、対応する特徴量を地図から検索でき、自己位置推

定が成功すると考えられる.

Table. 5 Success ratio of self-localization

	indoor	small outdoor	large outdoor
CIF method	57.5[%]	—	—
Bando's method[18](128)	80.8[%]	50.1[%]	46.4[%]
Bando's method[18](256)	86.7[%]	59.0[%]	52.6[%]
VAE($K=20$)	90.9[%]	96.0[%]	85.8[%]
VAE($K=50$)	90.3[%]	98.8[%]	87.0[%]
VAE($K=120$)	92.2[%]	98.6[%]	89.2[%]
LaserVAE($K=20$)	90.6[%]	96.0[%]	86.5[%]
LaserVAE($K=50$)	90.3[%]	98.6[%]	87.2[%]
LaserVAE($K=120$)	92.2[%]	98.4[%]	89.4[%]

VAE や LaserVAE は特徴量の次元数を変化させても, 地図内に登録されている特徴量から現在のスキャンに対応する特徴量の検索が安定して行え, 自己位置推定できていることが分かる. このことから, LaserVAE はスキャンから走行環境毎の不変的な構造を反映した特徴量を自動的に生成できていると考えられる.

3.3.4 環境地図のデータ量圧縮の性能

Fig.33 に各実験環境での提案手法と占有格子地図の地図のデータ量を示す. 提案手法のデータ量は学習済みの LaserVAE のニューラルネットワークを表すためのデータ量と, 特徴量と自己位置のペアで構成される特徴地図のデータ量とを合わせたものである. 従来から幅広く使われている占有格子地図は環境が大きくなるにつれデータ量の増加率が大きくなるのに対して, 提案手法はデータ量の増加率は小さい.

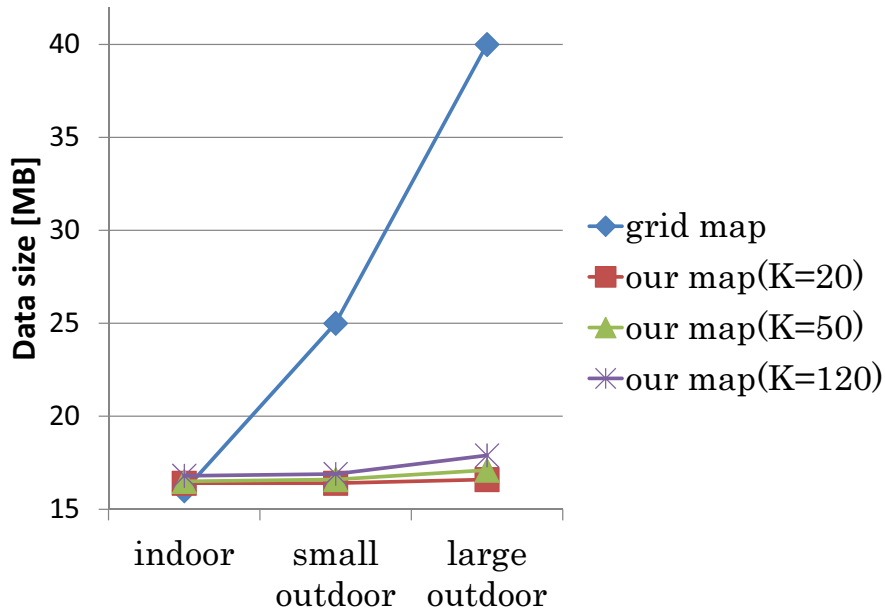


Fig. 33 Comparison of the size of map data

3.4 まとめ

レーザスキャナによって取得したスキャンから、走行環境を表現する特徴量を生成するために LaserVAE を提案した。

LaserVAE で自動的に得た特徴量で構成される特徴地図から、各特徴量についてスキャンを再構成し、紐づいている自己位置にスキャンを設置して作成した点群地図は、もとの地形が再現できることを確認した。また、実際の環境での大域的自己位置推定では、LaserVAE によって獲得した特徴量を用いた地図から、現在のスキャンに対応する特徴量を検索可能で、再構成したスキャンと現在のスキャンに ICP アルゴリズムを適用することで自己位置を正しく求めることができることを確認した。

提案手法は、生成する特徴量の次元数をパラメータとして持つが、特徴量の次元数を変化させても、自己位置推定の性能への影響が少ないことを確認した。これらのことから、提案手法 LaserVAE は、環境毎のスキャンから自己位置推定に有用な特徴量を生成できていると言える。

また、地図のデータ量に関しては、LaserVAE のネットワークと特徴量と自己位置の組を地図として保持することで、走行環境が広がっても、データ量の増加率は占有格子地図と比べて低いことを確認した。

今回構築した LaserVAE のネットワーク構造は、今回実験を行った環境で正しい答えが得られるよう調整されたものであり、ネットワーク構造が冗長になっている可能性がある。ネットワーク構造の層数や素子数をさらに調整することで、さらに、ネットワークのデータ量を減らせる可能性がある。また、

様々な環境において，今回構築したネットワーク構造が最適である保証はない．環境に応じて，ネットワーク構造の層数や素子数を自動的に決定できるような仕組みを考えることが今後の課題の1つである．

提案手法は，一度走行したコースのスキャンを用いて学習・地図作成を行っているため，走行するコースが障害物などで塞がれ，想定したコースを大きく外れる場合，自己位置推定がうまくいかなくなることが予想される．対処法としては，外れるコースの候補をあらかじめ地図として保持しておくことが考えられる．

4 結論

移動ロボットが環境内を自律的に行動するためには、自己位置推定技術が必要不可欠である。大域的な自己位置推定を行う手法として、特徴量を利用したスキャンマッチング法がある。スキャンマッチング法において、スキャン取得時に位置合わせに有効な特徴量の不足する場合があることや、ロボットの走行する環境に対応した特徴量を生成する必要があるなどの課題がある。本研究では、それぞれの課題に対して、選択的統合処理に基づく CIF ベース大域的自己位置推定 (提案手法 1), LaserVAE による特徴量生成とその特徴量に基づいた大域的自己位置推定 (提案手法 2) という手法を提案した。

第 1 章では、自己位置推定法の従来方法としてレーザスキャナから得られるスキャンと地図を重ね合わせるスキャンマッチング法を利用した方法があることを述べた。その手法における問題点としてスキャン内に位置合わせに有用な特徴の不足する状況では、位置合わせが不安定になることを述べた。これに対する従来方法として過去のスキャンを用いた局所地図を作成し、その局所地図と地図とのマッチングを行い、自己位置を推定するものがあり、その手法では、局所地図作成に用いるスキャン数が固定であることが問題であると述べた。また、人手によって設計された特徴量は環境の必ずしも環境に対応しているとは限らないため、環境に対応した特徴量を自動で生成することが望まれることを述べた。

提案手法 1 と提案手法 2 は特徴量を利用して大域自己位置推定を行う上で重要となるセンサ情報から地形の形状特徴を抽出する方法について、異なるアプローチを行った。

第 2 章 (提案手法 1) では、スキャン内に位置合わせに有用な特徴の不足する問題に対して、1 時刻分のスキャンでは位置合わせのための特徴が不足する場合の解決策として、過去のスキャンを利用し、局所地図を作成することで、特徴不足を補うことを提案した。局所地図を作成するためにスキャン内の位置合わせに有効な特徴が十分存在するか判別する識別器を作成した。判別には地形複雑度を用いた。スキャンから地形複雑度を算出し、作成した識別器によって局所地図作成が必要か判断した。スキャンマッチングには我々の研究グループで提案した CIF 特徴量を用いた。CIF ベーススキャンマッチングによって局所地図作成と大域的自己位置推定を行った。実験では、1 時刻分のスキャンでは自己位置推定に失敗するような状況であっても、提案手法によって作成した局所地図を用いることで自己位置推定に成功するようになることを示した。

第 3 章 (提案手法 2) では、スキャンから環境毎に特徴量を自動で生成するために “LaserVAE” を提案した。ステップエッジの含まれるようなスキャンデータも再構成できるように従来の VAE に step-edge classifier というネットワークを追加した。LaserVAE によってスキャンから生成した特徴量を用いて地図を作成しておき、その特徴量の対応をとることで粗な位置を推定した後、ICP アルゴリズムを用い

ることで詳細に自己位置推定を行うことを述べた。実験では、屋内・屋外の人の往来のある 3 つの環境を用意し自己位置を行った。LaserVAE によってステップエッジを含むようなスキャンも再構成できることを示した。また、人の往来のある環境においても自己位置推定を行うことができ、地図のデータ量も少なくなることを示した。

提案手法 1 の特徴的な点は、スキャンマッチングの成否を予め予測することである。こうすることで、特徴が不足する時のみ局所地図を作成することができ、局所地図作成に用いるスキャンを最小にできる。

この手法の課題として、未学習の環境では、識別器によるスキャンマッチングの成否の判別がうまくいかない可能性がある。解決策として、予めあらゆる環境における地形複雑度とスキャンマッチングの成否を取得して識別器を学習しておくことが考えられる。また、過去のスキャンをどれだけ用いても位置合わせに有効な特徴が不十分だった場合に、スキャンマッチングに失敗することが予想される。この問題は、レーザスキャナの設置された高さに形状特徴が存在しないという問題であるので、レーザスキャナの高さを変更する・回転させるあるいは他のセンサを使用するなどしてより多くの環境情報を取得することで解決可能であると考ええる。

提案手法 2 の特徴的な点は、2D スキャンのような粗なデータに対して圧縮・復元できるような学習を行っている点である。地図として、特徴量 (圧縮したスキャン) のみを用いることで、地図のデータ量を少なくすることができる。この手法の課題として、走行するコースが大きく変更された場合、位置合わせがうまくいかないことが予想される。解決策の一つとして、複数のコース・位置姿勢を地図に追加することが考えられる。今後の展望として、2D スキャンデータだけでなく 3D スキャンデータに応用することが考えられる。3D スキャンデータは 2D スキャンデータに比べて処理するデータ量が膨大になるため、提案手法 2 で提案したようにスキャンデータを特徴量として低次元のベクトルに変換することが有効となると考えられる。

本論文における提案手法 1 と 2 の位置づけ

提案手法 1 においてスキャンマッチングする際には、CIF 特徴量に基づいて対応点を探索している。CIF 特徴量はキーポイント周辺の点群の分布を用いて計算されるため、回転不変性 (スキャンデータ全体が回転しても、回転の前後で、スキャンデータ中のある点、すなわち、キーポイントにおける CIF 特徴量は同じになる) を持つため、オフィスのような屋内環境では、不連続点が多少観測されたとしても、スキャンデータが時間的に安定している部分から、一致する箇所を発見し自己位置推定をすることが可能である。

しかしながら、スキャンデータが時間的に不安定な環境、例えば、ガラスの壁を2次元レーザスキャナで計測するような状況では、特徴量の一意性（スキャンデータ中の1点と実際の環境における場所が対応している状況）が保証されない。そのため、屋外の環境などのようにオクルージョンなどによってスキャンデータが時間的に安定して観測できない状況では、自己位置推定に有用な特徴が十分に存在する局所地図を生成できない可能性が高くなり、自己位置推定を行うことは困難であると考えられる。

従って、提案手法1は、屋外環境のようなオクルージョンが頻発するような環境では、CIF特徴量の一意性が保証されず、自己位置推定に失敗する可能性が高くなる。また、地図が大きくなると現在のスキャンと地図との対応点を探索するときに対応するキーポイントの候補数が膨大になり、計算時間がそれに比例して増加する。

これらの理由から、提案手法1は、スキャンデータが時間的に不安定な環境や、大規模な環境で用いるのには向いていないと考えられる。一方で、提案手法1は、スキャンデータが時間的に安定な環境で、小規模な環境では、最も高いパフォーマンスを発揮することができると考えられる。

提案手法2は、各環境で取得した大量のスキャンデータを、VAEによりデータ圧縮することにより特徴量を生成している。VAEのデータ圧縮機能によって得られる特徴量は、その生成過程において、大量のスキャンデータに共有されるような、つまり、定常的にスキャンデータ内に存在するような特徴量が得られるため、時間的に安定して観測されないようなスキャンデータからは特徴量は生成されない。従って、スキャンデータ内の一部に時間的に安定して観測されないデータ、例えば、人の足に相当するようなデータが含まれている場合でも、その他のスキャンデータ内に含まれている時間的に安定して観測されるデータから特徴量を生成することができるため、提案手法2は実験結果に示したように、動的な環境において使用可能であると考えている。また、VAEのネットワーク構造により、大量のスキャンデータが持つ情報を圧縮して保持できることから、ロボットの走行環境が広範囲になっても、VAEのencoderによって生成され低次元の特徴ベクトルによって環境の地図情報を保持することが可能である。

しかし、実験結果で示したように、VAEにより生成される特徴量には回転不変性がないため、何らかの理由（例えば、走行コース上に障害物があり、それを回避するような状況）で地図作成時の走行コースから外れて移動する場合、ロボット自体が回避のために姿勢を大きく変えなければならず、その結果として自己位置推定に失敗することが予想される。同様な状況として、地図作成時と同じ走行コース上で逆向きに走行するような状況においても、時々刻々と入力されるスキャンと地図作成時のスキャンは、観測される環境中の範囲が異なる。このように、観測される環境中の範囲が異なると、同じ場所において取得したスキャンであっても、取得されるスキャンデータは大きく異なる（1回のスキャンの

計測範囲は 270 度であるので、同じ場所でレーザースキャナの向きが 180 度異なると、レーザースキャナの共通の計測範囲は 180 度になる) ため、2 つのスキャン内に共通するデータが少なり、地図データと一致する特徴量が生成されず、自己位置推定に失敗すると考えられる。

これらの理由から、提案手法 2 は、地図作成時とほぼ同じコース上を走行する場合においては、人の往来のあるような動的な環境であっても、自己位置推定が可能であり、走行環境のサイズが大きくなっても地図データ量を抑えることが可能である。

しかし、何らかの理由で地図生成時の走行コースから外れる状況では、自己位置推定に失敗するため、そのような状況下では、提案手法 1 のような局所的地図を生成して地図データと一致する箇所を探索する手法を用いて自己位置推定を行うことが必要であると考えられる。

本研究は 2D スキャンデータを用いた自己位置推定の課題に対して 2 つのアプローチを行い、問題解決を目指した。基本的には提案手法 2 を用いて自己位置推定を行い、提案手法 2 が失敗するような状況で局所的に提案手法 1 を用いて自己位置推定を行えば、様々な状況下で自己位置推定が頑健に行えるようになると考えられる。

本研究の成果が今後の自己位置推定法の発展の一助になれば幸いである。

謝辞

本論文をまとめるにあたって，種々のご指導を頂き，格別の配慮を賜った和歌山大学大学院システム工学研究科の中村恭之教授，中嶋秀朗教授，長瀬賢二教授，八谷大岳講師に心から御礼申し上げます．また，日常の議論を通じて多くの知識や示唆を頂いた中村研究室の皆様には感謝いたします．

参考文献

- [1] J. Zhang and S. Singh: "LOAM: Lidar Odometry and Mapping in Real-time," Proc. of RSS2014, 2014.
- [2] J. Zhang and S. Singh: "Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast," ICRA2015, pp.2174–2181, 2015.
- [3] P. J. Besl and N. D. McKay: "A Method for Registration of 3-D Shapes," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol.14, No.2, pp.239–256, 1992.
- [4] F. Lu and E. Milios: "Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans," Journal of Intelligent and Robotic Systems, vol.18, pp.249–275, 1997.
- [5] P. Biber, and W. Strasser: "The normal distributions transform: a new approach to laser scan matching," Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), pp.2743–2748, 2003.
- [6] J.S. Gutmann, T. Weigel and B. Nebel: "Fast, Accurate, and Robust Self-Localization in Polygonal Environments," *Proc. of IROS'99*, pp.1412–1419, 1999.
- [7] P. Jensfelt and S. Kristensen: "Active Global Localization for a Mobile Robot Using Multiple Hypothesis Tracking," *IEEE Trans. on Robotics and Automation*, Vol.17, No.5, pp. 748–760, 2001.
- [8] 友納 正裕: "ユークリッド変換に不変な特徴量を用いた二次元大域スキャンマッチング方式," 日本ロボット学会誌, Vol.25, No.03, pp.66–77, 2007.
- [9] 中村 恭之, 脇田 翔平: "2D スキャンデータの合同変換に不変な特徴量 (CIF) を用いたロバストなスキャンマッチング," 第 19 回ロボティクスシンポジウム, 6C3, pp.592–598, 2014.
- [10] 中村 恭之, 脇田 翔平: "合同変換に不変な特徴量 (CIF) とキーポイント間の幾何学的拘束に基づいたロバストなスキャンマッチング法の提案," 計測自動制御学会論文集 Vol.51, No.5, pp.309–318, 2015.
- [11] 脇田 翔平, 中村 恭之: "スキャンデータの順序保存統合を用いた CIF ベーススキャンマッチング法," ロボティクス・メカトロニクス講演会 '15, 2A2-M09, 2015.
- [12] F. Dellaert, D. Fox, W. Burgard, and S. Thrun: "Monte Carlo Localization for Mobile Robots," *Proc. ICRA '99*, pp.1322–1328, 1999.
- [13] 友納正裕: "2D レーザスキャナによる SLAM における地図の点群表現とループ閉じ込み," 第 22

回ロボティクスシンポジア, pp.42–47, 2017.

- [14] S. Thrun, W. Burgard and D. Fox: "Probabilistic ROBOTICS," MIT Press, 2005.
- [15] G. Giorgio, C. Stachniss, and W. Burgard: "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE transactions on Robotics* 23.1, pp.34–46, 2007.
- [16] Y. Li and E. Olson: "Extracting general-purpose features from LIDAR data," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010.
- [17] Y. Li and E. Olson: "Structure tensors for general purpose lidar feature extraction," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1869–1874, 2011.
- [18] 阪東茂, 原祥亮, 坪内孝司: "2次元測域センサデータの空間周波数解析に基づく大域的自己位置推定," 第18回ロボティクスシンポジア, pp.163–168, 2013.
- [19] G. D. Tipaldi and K. O. Arras: "Flirt-interest regions for 2d range data," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 3616–3622, 2010.
- [20] F. Kallasi, D. L. Rizzini and S. Caselli: "Fast Keypoint Features from Laser Scanner for Robot Localization and Mapping" *IEEE Robotics and Automation Letters*, 2016.
- [21] 脇田翔平, 中村恭之: "選択的統合処理に基づく CIF ベース大域的自己位置推定," *日本ロボット学会誌*, Vol.63, No.06, pp.41–50, 2018.
- [22] S. Zhao, J. Song and S. Ermon: "InfoVAE: Information Maximizing Variational Autoencoders," *arxiv*, 2017.
- [23] X. Hou, L. Shen, K. Sun and G. Qiu: "Deep Feature Consistent Variational Autoencoder," *arXiv*, 2016.
- [24] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua: "StyleBank: An Explicit Representation for Neural Image Style Transfer," *CVPR*, 2017.
- [25] K. Konolige, and K. Chou: "Markov localization using correlation," *Proc. IJCAI'99*, pp.1154–1159, 1999.
- [26] E. Olson: "Real-time correlative scan matching," *Proc. of ICRA '09*, 2009.
- [27] J. Gutmann, and D. Fox: "An Experimental Comparison of Localization Methods Continued," in *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems(IROS' 02)*, pp.454–459, 2002.
- [28] 上田隆一, 新井民夫, 浅沼和範, 梅田和昇, 大隅久: "パーティクルフィルタを利用した自己位置推定に生じる致命的な推定誤りからの回復法," *日本ロボット学会誌*, Vol.23, No.4, pp.466–473, 2005.

- [29] A. Diosi, L. Kleeman: "Laser scan matching in polar coordinates with application to SLAM," *Proc. of IROS'05*, pp.3317–3322, 2005.
- [30] 藤井祐介, 大矢晃久, 坪内孝司: "人混み中での移動ロボットの自己位置推定に有効な測域センサデータ処理法の提案 -蓄積処理と近傍点削除による壁面の抽出-", ロボティクス・メカトロニクス講演会'14, 2A2-Q01, 2014.
- [31] 田下 裕一, 中村 恭之: "ロバストなスキャンマッチングのための合同変換に不変な特徴量 (CIF) の提案," ロボティクス・メカトロニクス講演会'13, 1A2-H10, 2013.
- [32] 原 祥堯, 坪内 孝司, 大島 章: "確率的に蓄積したスキャン形状により過去を考慮した Rao-Blackwellized Particle Filter SLAM," 日本機械学会論文集, Vol.82, No.834 p.15-00421, 2016.
- [33] V.N., Vapnik: "Statistical Learning Theory," *J. Wiley*, 1998.
- [34] V.N., Vapnik: "The Nature of Statistical Learning Theory (Second Edition)," *Springer*, 1999.
- [35] D. G. Lowe: "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol.60, no.2, pp.91–110, November 2004.
- [36] 友納正裕: "観測度数をもつ占有格子地図による長時間地図構築のためのデータ削減手法" 日本機械学会論文集, 2018.
- [37] G. A. Borges and M. J. Aldon: "Line extraction in 2D range images for mobile robotics," *Journal of Intelligent and Robotic Systems*, Vol.40, pp.267–297, 2004.
- [38] P.J. Besl and N.D. McKay: "A Method for Registration of 3-D Shapes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol.14, No.2, pp.239–256, 1992.
- [39] K. Lingemann, H. Surmann, A. Nuchter, and J. Hertzberg: "Indoor and outdoor localization for fast mobile robots," *Proc. of IROS'04*, pp.2185–2190, 2004.
- [40] 中村 恭之: "制約条件付き最小 2 乗法に基づく高速な局所型スキャンマッチング法," 日本ロボット学会誌, Vol.28, No.5, pp.648–657, 2010.

研究業績

論文誌

- [1] 脇田翔平, 中村恭之, 八谷大岳: “LaserVAE による特徴量生成とその特徴量に基づいた大域自己位置推定”, 計測自動制御学会論文集, Vol.55, No.7, pp.476–483, 2019.
- [2] 脇田翔平, 中村恭之: “選択的統合処理に基づく CIF ベース大域的自己位置推定 “, 日本ロボット学会誌, Vol.63, No.06, pp.41–50, 2018.
- [3] 中村恭之, 脇田翔平: “合同変換に不変な特徴量 (CIF) とキーポイント間の幾何学的拘束に基づいた ロバストなスキャンマッチング法の提案”, 計測自動制御学会論文集 Vol. 51, No. 5, pp.309–318, 2015.

国際会議

- [1] S. Wakita, T. Nakamura, H. Hachiya: “Laser variational autoencoder for map construction and self-localization” , Proceedings of 2018 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp.3983–3988, 7-10 Oct. 2018.
- [2] T. Nakamura and S. Wakita, “Robust global scan matching method using congruence transformation invariant feature descriptors and a geometric constraint between keypoints” , Proceedings of 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), pp.3103–3108, 5-8 Oct. 2014.

国内会議 (査読有)

- [1] 脇田翔平, 中村恭之: “スキャンデータの順序保存統合と統合処理判断機能を持つ CIF ベーススキャンマッチング法,” 第 21 回ロボティクスシンポジア, 3A1, pp.172–177, 2016.
- [2] 中村恭之, 脇田翔平: ” 2D スキャンデータの合同変換に不変な特徴量 (CIF) を用いたスキャンマッチング” , 第 19 回ロボティクスシンポジア, 6C3, pp.592–598, 2014.

国内会議 (査読無)

- [1] 脇田翔平, 中村恭之, 八谷大岳: “オートエンコーダを用いた環境地図の特徴表現と自己位置推定,” ロボティクス・メカトロニクス講演会’ 18, 1P1-G06, 2018.
- [2] 脇田翔平, 中村恭之: “スキャンデータの順序保存統合を用いた CIF ベーススキャンマッチング法,” ロボティクス・メカトロニクス講演会’ 15, 2A2-M09, 2015.
- [3] 脇田翔平, 中村恭之: “合同変換に不変な特徴量 (CIF) とキーポイントの幾何学的拘束に基づいたロバストなスキャンマッチング法の提案,” ロボティクス・メカトロニクス講演会’ 14, 2A2-T05, 2014.
- [4] 中村恭之, 脇田翔平: “2D スキャンデータの合同変換に不変な特徴量 (CIF) を用いたロバストなスキャンマッチング法,” 日本ロボット学術講演会 2013, (3J1-02), 2013.